

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
FACULDADE DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**QOC \*: utilizando *Design Rationale* como  
ferramenta para Gestão do Conhecimento**

Marcos Camponogara

Orientadora: Prof. Dr. Milene Selbach Silveira

**Dissertação de Mestrado**

Porto Alegre  
2008

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL  
FACULDADE DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MARCOS CAMPONOGARA

**QOC \*: utilizando *Design Rationale* como  
ferramenta para Gestão do Conhecimento**

Dissertação apresentada como requisito parcial à  
obtenção do grau de Mestre, pelo programa de Pós  
Graduação em Ciência da Computação da  
Pontifícia Universidade Católica do Rio Grande do  
Sul.

Orientadora: Prof. Dr. Milene Selbach Silveira

Porto Alegre  
2008

### Dados Internacionais de Catalogação na Publicação (CIP)

C198Q Camponogara, Marcos  
QOC\* : utilizando *Design Rationale* como ferramenta para  
gestão do conhecimento / Marcos Camponogara. – Porto Alegre,  
2008.  
87 f.  
  
Diss. (Mestrado) – Fac. de Informática, PUCRS  
Orientador: Profa. Dra. Milene Selbach Silveira  
  
1. Informática. 2. Engenharia de Software. 3. Gestão do  
Conhecimento. 4. Design (Informática). 5. Tomada de  
Decisões. I. Silveira, Milene Selbach. II. Título.  
  
CDD 005.1

Ficha Catalográfica elaborada pelo  
Setor de Tratamento da Informação da BC-PUCRS



Pontifícia Universidade Católica do Rio Grande do Sul  
FACULDADE DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

## TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "**QOC\*: Utilizando *Design Rationale* como Ferramenta para Gestão do Conhecimento**", apresentada por Marcos Camponogara, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Inteligência Artificial, aprovada em 09/01/08 pela Comissão Examinadora:

\_\_\_\_\_  
Profa. Dra. Milene Selbach Silveira -  
Orientadora

PPGCC/PUCRS

\_\_\_\_\_  
Prof. Dr. Marcelo Blois Ribeiro

PPGCC/PUCRS

\_\_\_\_\_  
Prof. Dr. Michael da Costa Mora -

FACIN/PUCRS

Homologada em 20.../05.../08..., conforme Ata No. 010/08. pela Comissão Coordenadora.

\_\_\_\_\_  
Prof. Dr. Fernando Gehm Moraes  
Coordenador.



PUCRS

**Campus Central**

Av. Ipiranga, 6681 - P32 - sala 507 - CEP: 90619-900

Fone: (51) 3320-3611 - Fax (51) 3320-3621

E-mail: [ppgcc@inf.pucrs.br](mailto:ppgcc@inf.pucrs.br)

[www.pucrs.br/facinf/pos](http://www.pucrs.br/facinf/pos)

*Knowledge is power.*

Francis Bacon

## **AGRADECIMENTOS**

À minha orientadora, Prof. Dr. Milene, pela dedicação, empenho e contribuição durante o desenvolvimento desta dissertação.

À minha esposa Carolina, pelo apoio, pelas palavras de incentivo perante as dificuldades e pela compreensão de minhas ausências.

Aos meus pais e ao meu irmão pelo apoio incondicional e pelo conforto em todos os momentos.

## RESUMO

Durante o processo de desenvolvimento de um sistema de software, uma grande quantidade de conhecimento é utilizada e produzida como resultado das opções analisadas e das decisões tomadas ao longo do desenvolvimento do projeto. Este conhecimento é valioso, pois reflete as razões que estão por trás das decisões, o que facilita o entendimento dos rumos do projeto e proporciona uma visão global do mesmo. Desta forma, existe a necessidade de se encontrar alternativas para organizar e manter este tipo de conhecimento e então torná-lo um recurso que possa facilitar a continuidade de projetos de software ou então a manutenção de sistemas desenvolvidos. Outra possibilidade ainda é a reutilização deste conhecimento em outros projetos, considerando-se neste caso a existência de cenários semelhantes aos cenários vivenciados em projetos anteriores.

Neste sentido, o presente trabalho aborda o desenvolvimento de uma pesquisa baseada em Gestão do Conhecimento e *Design Rationale*, e propõe uma maneira de representar e manter as razões que motivaram a tomada de determinadas decisões em projetos de software, considerando, para isso, uma forma de representação simples e objetiva que instiga o questionamento e a discussão a respeito das melhores opções para atender questões que surgem durante o desenvolvimento de um projeto.

**Palavras-chave:** Gestão do Conhecimento. *Design Rationale*. Tomada de decisão.

## **ABSTRACT**

In general, the decision making process in a software development project depends on the people involved in this process and their knowledge of the project. Also, the option analysis process which results in a decision, produces a kind of knowledge that is very important for the project and its future, since this knowledge is about key decisions and their rationale. Thus, understanding the rationale behind a decision provides a powerful way to better understand how a project works in detail and why some decisions were taken. Considering this perspective, it is necessary to find an approach to organize and maintain knowledge about the decision making process from software development projects, and then use this knowledge to help people involved in these projects. Another possibility is to reuse the generated knowledge in other projects, as learned lessons, considering in this case, similar scenarios from past projects.

Taking in account what has been stated above, this research proposes an approach to document and organize the rationale regarding the decision making process in software development projects, using for this purpose a simple and objective representation that encourages reasoning in order to define the best solutions for problems faced in software development projects.

**Key-words:** Knowledge Management. Design Rationale. Decision Making.

## LISTA DE FIGURAS

<b>Figura 2.1:</b> Processos de conversão do conhecimento segundo Nonaka (1991) .....	18
<b>Figura 3.1:</b> Exemplo genérico do mapa de temas [MEDEIROS, 2006].....	35
<b>Figura 3.2:</b> Exemplo genérico da notação QOC, adaptado de Shum e Hammond (1994).....	37
<b>Figura 3.3:</b> Vocabulário da linguagem DRL [MEDEIROS, 2006]. .....	39
<b>Figura 4.1:</b> Modelagem das decisões utilizando QOC.....	47
<b>Figura 4.2:</b> Modelagem das decisões utilizando DRL.....	47
<b>Figura 5.1:</b> Exemplo conceitual utilizando a representação de importância das questões. ....	57
<b>Figura 5.2:</b> Exemplo conceitual utilizando a representação de pesos para os critérios. ....	58
<b>Figura 5.3:</b> Exemplo conceitual com a representação que destaca a opção escolhida. ....	68

## LISTA DE TABELAS

<b>Tabela 5.1:</b> Sugestão de tabela para representação textual das principais Questões e Critérios. ....	59
<b>Tabela 5.2:</b> Tempo de experiência dos participantes do experimento por área.....	60
<b>Tabela 5.3:</b> Informações complementares sobre o perfil dos participantes do experimento. ....	61
<b>Tabela 5.4:</b> Resultados do experimento.....	63
<b>Tabela 5.5:</b> Sugestão de tabela para representação textual das Questões, Opções e Critérios.....	66

## LISTA DE SIGLAS

CMMI – *Capability Maturity Model Integration*

DRL – *Decision Representation Language*

IBIS – *Issue-Based Information System*

QOC – *Questions, Options and Criteria*

PHI – *Procedural Hierarchy of Issues*

# SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>14</b>
<b>2 GESTÃO DO CONHECIMENTO</b> .....	<b>16</b>
<b>2.1 Conhecimento</b> .....	<b>16</b>
<b>2.2 Tipos de Conhecimento</b> .....	<b>17</b>
<b>2.3 Definições de Gestão do Conhecimento</b> .....	<b>19</b>
<b>2.4 Vantagens da Gestão do Conhecimento</b> .....	<b>21</b>
<b>2.5 Fatores Chave da Gestão do Conhecimento</b> .....	<b>22</b>
<b>2.6 Projetos de Gestão do Conhecimento</b> .....	<b>24</b>
<b>2.6.1 Cultura Orientada para o Conhecimento</b> .....	<b>24</b>
<b>2.6.2 Infra-estrutura Técnica e Organizacional</b> .....	<b>25</b>
<b>2.6.3 Apoio Gerencial</b> .....	<b>26</b>
<b>2.6.4 Vinculação ao Valor Econômico e Setorial</b> .....	<b>26</b>
<b>2.6.5 Clareza de Visão e Linguagem</b> .....	<b>27</b>
<b>2.6.6 Elementos Motivadores</b> .....	<b>27</b>
<b>2.6.7 Estruturação do Conhecimento</b> .....	<b>28</b>
<b>2.6.8 Múltiplos Canais para Transferência do Conhecimento</b> .....	<b>28</b>
<b>2.6.9 Avaliação</b> .....	<b>29</b>
<b>2.7 Gestão do Conhecimento e Desenvolvimento de Software</b> .....	<b>29</b>
<b>2.7.1 Capital Intelectual</b> .....	<b>29</b>
<b>2.7.2 Compartilhamento do Conhecimento</b> .....	<b>30</b>
<b>2.7.3 Aquisição do Conhecimento</b> .....	<b>31</b>
<b>3 DESIGN RATIONALE</b> .....	<b>33</b>
<b>3.1 Abordagens de Design Rationale</b> .....	<b>34</b>
<b>3.1.1 IBIS</b> .....	<b>34</b>
<b>3.1.2 QOC</b> .....	<b>36</b>
<b>3.1.3 DRL</b> .....	<b>38</b>
<b>3.2 Requisitos para a Representação de Design Rationale</b> .....	<b>39</b>
<b>4 ESTUDOS PRELIMINARES</b> .....	<b>41</b>
<b>4.1 Caracterização do Problema</b> .....	<b>41</b>
<b>4.2 Escopo do Trabalho</b> .....	<b>42</b>
<b>4.3 Estudo de Caso</b> .....	<b>43</b>
<b>4.3.1 Descrição do Sistema Objeto</b> .....	<b>43</b>

4.3.2 Principais Problemas de Decisão Envolvidos no Projeto .....	45
4.3.3 Modelagem das Decisões.....	46
4.3.3.1 QOC.....	46
4.3.3.2 DRL .....	46
4.3.3.3 Modelagem Textual .....	48
4.3.4 Entrevistas.....	50
4.3.4.1 Perfil dos Entrevistados .....	50
4.3.4.2 Dinâmica das Entrevistas .....	50
4.3.4.3 Análise das Entrevistas .....	51
4.3.4.3.1 Observações Gerais.....	51
4.3.4.3.2 Observações Específicas .....	52
4.3.4.3.3 Sugestões .....	53
5 QOC * .....	54
5.1 Características do QOC * .....	54
5.1.1 Notação Simples .....	55
5.1.2 Representação da Importância das Questões .....	56
5.1.3 Representação do Peso dos Critérios .....	57
5.1.4 Descrição Textual .....	58
5.2 Aplicação da Proposta.....	59
5.3 Discussão .....	63
5.4 Refinamento da Proposta.....	66
5.4.1 Ampliação da Descrição Textual.....	66
5.4.2 Tipos de Decisão.....	67
5.4.3 Identificação da Decisão Tomada .....	67
6 CONSIDERAÇÕES FINAIS .....	69
6.1 Conclusões.....	69
6.2 Trabalhos Futuros.....	71
REFERÊNCIAS .....	72
APÊNDICE A.....	76
APÊNDICE B.....	77
APÊNDICE C.....	78
APÊNDICE D.....	87

# 1 INTRODUÇÃO

Dentro do contexto de um projeto de software, a maioria das atividades envolve algum tipo de conhecimento, principalmente as decisões de projeto que são tomadas com base em conhecimento prévio sobre o assunto em questão. O fato de se ter um maior conhecimento sobre uma área específica pode levar a uma decisão mais acertada e, conseqüentemente, gerar melhores resultados.

De maneira geral, o conhecimento se desenvolve ao longo do tempo, a partir de experiências variadas, que abrangem a aprendizagem formal (livros, cursos, professores) e também informal, através da prática e da troca de experiências. Além disto, diante de situações que exijam soluções novas e criativas, novos conhecimentos podem ser criados a partir da associação de conhecimentos pré-existentes [DAVENPORT e PRUSAK, 1998].

Projetos de software, em geral, possuem uma forte ligação com o conhecimento, sendo que, neste contexto, a Gestão do Conhecimento permite um melhor aproveitamento do capital intelectual e do conhecimento existentes nestes projetos e, principalmente, oferece subsídios para o aprimoramento do conhecimento e do aprendizado a partir das experiências das pessoas envolvidas.

Além disso, neste tipo de projeto, uma série de decisões-chave é tomada com o objetivo de melhor atender aos requisitos do software que será desenvolvido. Porém, em grande parte das vezes, o porquê destas decisões acaba se perdendo e, conseqüentemente, dificultando a continuidade do projeto ou a manutenção do software, principalmente se estas tarefas forem desempenhadas por pessoas que não participaram do projeto desde o início.

Uma opção para tentar resolver os problemas relacionados à falta de informação sobre essas decisões, são as abordagens de *Design Rationale*, as quais buscam manter as decisões tomadas durante o desenvolvimento de um projeto, mantendo também suas justificativas (o porquê destas decisões) [MORAN e CARROLL, 1996; LEE, 1997]. O objetivo destas abordagens é possibilitar que não se perca tempo com alternativas de projeto já avaliadas anteriormente, e que não puderam ser aplicadas, e, mais do que isso, se espera que, a captura do *rationale* de um projeto, possibilite a reutilização das decisões e experiências em outros projetos.

O presente trabalho busca utilizar o *Design Rationale* como um meio para realizar Gestão do Conhecimento, e assim propor uma forma para documentar e manter as opções de projeto discutidas e suas razões, bem como para identificar as decisões tomadas. Acredita-se que este tipo de documentação seja bastante útil em projetos de software, seja futuramente no mesmo projeto ou então em outros projetos, como lições aprendidas.

O objetivo deste trabalho é especificar uma técnica baseada em *Design Rationale* que possa ser utilizada para gerenciar o conhecimento relacionado ao processo de tomada de decisão em projetos de software. Através desta técnica, pretende-se representar os problemas e alternativas ponderadas antes de cada decisão de forma simples e intuitiva, reduzir a perda de capital intelectual quando pessoas saem do projeto, propiciar a outros projetos o reuso de experiências positivas e prevenir experiências negativas, ambas motivadas por decisões de projeto.

O presente trabalho está organizado da seguinte forma: os capítulos 2 e 3 abordam os principais conceitos envolvidos no desenvolvimento desta dissertação, o capítulo 4 apresenta os detalhes e os resultados do estudo de caso realizado para obter o embasamento para a proposta, sendo que a mesma é apresentada e detalhada no capítulo 5. Ainda no capítulo 5, são apresentados os detalhes e os resultados da aplicação da proposta em um projeto de software fictício. No capítulo 6 são feitas as conclusões do trabalho e são apresentadas algumas sugestões de trabalhos futuros.

## 2 GESTÃO DO CONHECIMENTO

A Gestão do Conhecimento é um tema que tem recebido grande atenção nos últimos tempos. Basicamente, isto se deve ao fato de vivermos, segundo Drucker (1997), no que pode ser chamada de “a sociedade do conhecimento”. Nesta sociedade, cada vez mais o conhecimento se torna importante para os processos produtivos. E, o aprimoramento e o compartilhamento do conhecimento são itens valiosos para a melhoria constante desses processos.

É neste contexto que a Gestão do Conhecimento surge, como ferramenta para auxiliar na identificação, formalização e disseminação do conhecimento e, com isso, proporcionar uma maior vantagem competitiva para as organizações. Em um mundo onde o acesso à informação é cada vez mais fácil, este tipo de ferramenta oferece uma forma de gerenciar, filtrar e destacar o conhecimento que é útil para o sucesso de determinado projeto, empresa ou organização.

Este capítulo aborda o conceito de conhecimento e como ele é criado, bem como os tipos de conhecimento que podem existir em uma organização. Além disso, trata da Gestão do Conhecimento em si, das vantagens de se ter um processo de gerência do conhecimento e, também, dos fatores chaves para a implantação de um projeto de Gestão do Conhecimento.

### 2.1 Conhecimento

O conhecimento é derivado de informações, informações são derivadas de dados e dados são um conjunto estruturado de transações sem qualquer significado inerente. Enquanto os dados são material bruto, fatos discretos, a informação tem a capacidade de modificar a percepção sobre determinado evento, consiste em dados que foram organizados e modelados com algum propósito [DAVENPORT e PRUSAK, 1998].

Segundo Hussain et. al. (2004), dados são coleções de fatos, medidas e estatísticas enquanto informações são dados organizados ou processados que possuem referência temporal e correte. Já o conhecimento, é caracterizado por informação contextual, relevante e fundamentada. O conhecimento é proveniente de elementos de reflexão e experiência que o

distinguem de informação em um determinado contexto. Diferente de informação, conhecimento pode ser utilizado para solucionar problemas.

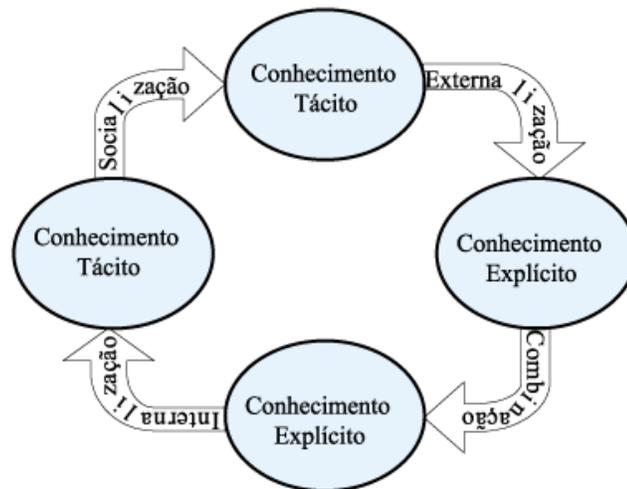
Este autor afirma, ainda, que, ao passo que dados, informações e conhecimento podem, de maneira geral, ser vistos como patrimônio de uma organização, o conhecimento proporciona significado aos dados e informações tornando-os muito mais valiosos. O conhecimento continua a ter valor e pode ser reutilizado mesmo com o passar do tempo além de ter relevância histórica, enquanto a informação tende a perder valor sem a preservação do contexto no qual foi adquirida. Em suma, informação pode ser acumulada com o passar do tempo, mas conhecimento pode ser agregado e aprimorado.

## 2.2 Tipos de Conhecimento

Segundo Nonaka (1991), existem dois tipos de conhecimento: o conhecimento **explícito** e o conhecimento **tácito**. O primeiro se caracteriza como o ponto final de um processo, ou seja, é formal, sistemático e, sobretudo, facilmente comunicado e compartilhado através de recursos como especificações, livros, fórmulas, etc. O segundo, porém, é um tipo de conhecimento que não se expressa facilmente, pois o mesmo é altamente pessoal e de difícil formalização.

Este autor afirma que o conhecimento tácito é composto por habilidades técnicas e pelo *know-how*, ou seja, pela experiência de determinado indivíduo em realizar certa atividade. Ao mesmo tempo, este tipo de conhecimento tem importante dimensão cognitiva, consistindo de modelos mentais, crenças e perspectivas bem fundamentadas que são tidas como certas. Por isso, estes modelos implícitos influenciam diretamente a maneira como o mundo é percebido pelas pessoas.

O autor propõe, ainda, que a distinção entre conhecimento tácito e explícito sugere quatro padrões básicos para a criação de conhecimento em qualquer organização (Figura 2.1).



**Figura 2.1:** Processos de conversão do conhecimento segundo Nonaka (1991)

Estes padrões são:

1. Socialização (de tácito para tácito): é o conhecimento compartilhado diretamente de uma pessoa para outra como, por exemplo, em reuniões nas quais experiências são expostas e discutidas. Neste caso, o conhecimento gerado é somente tácito, não há a criação de conhecimento explícito, ou seja, uma pessoa consegue absorver implicitamente o conhecimento proveniente de outra pessoa.
2. Combinação (de explícito para explícito): consiste na combinação de elementos isolados de conhecimento explícito para a criação de um novo conhecimento. Neste contexto, o conhecimento explícito pode ser compartilhado através de reuniões, documentos, e-mail, etc. Como exemplo, pode-se citar o compartilhamento de documentos e informações por várias pessoas ou a elaboração de relatórios que sintetizam informações provenientes de várias fontes diferentes.
3. Externalização (de tácito para explícito): a externalização é o processo de converter conhecimento tácito em explícito, apesar da dificuldade de explicitar-se este tipo de conhecimento. Basicamente, isto acontece quando um indivíduo consegue expressar seu conhecimento de forma que o mesmo possa ser capturado e, então, compartilhado explicitamente. Este processo tipicamente pode acontecer, por exemplo, através do diálogo entre pessoas e a partir de respostas formais a perguntas.

4. Internalização (de explícito para tácito): a partir do momento em que um novo conhecimento explícito passa a ser compartilhado, por exemplo, em uma organização, os indivíduos desta organização têm a oportunidade de internalizá-lo e, desta forma, ampliar e reformular seus próprios conhecimentos tácitos. Um exemplo pode ser a leitura de documentos ou livros provenientes de diferentes fontes, o que permite que um indivíduo aprenda o que outras pessoas aprenderam anteriormente e crie um conhecimento tácito através da combinação de seu conhecimento com o conhecimento de outros.

Holsapple and Whiston (1996) e Zack (1999), porém, definem a existência de seis tipos de conhecimento: **descritivo**; **procedural**; **racional**; **lingüístico**; **representativo**; e, **assimilativo**.

Para os autores, o conhecimento **descritivo** está relacionado a informações sobre o passado, o presente, o futuro ou estados hipotéticos de relevância, e tem como objetivo responder “o que é”. O conhecimento **procedural** busca entender o “como” e especifica os passos para realizar determinada tarefa. O conhecimento **racional** questiona o “por que”, resultando em conclusões válidas para determinado conjunto de circunstâncias.

Já o conhecimento **representativo** facilita o processo de comunicação e se preocupa com a maneira que o conhecimento é transmitido. O conhecimento **lingüístico** interpreta o conhecimento transmitido e o conhecimento **assimilativo** ajuda a manter a base de conhecimento através da incorporação de novos conhecimentos. Os três primeiros tipos de conhecimento (descritivo, procedural e racional) são os conhecimentos básicos que uma organização possui e que a permitem realizar suas tarefas, já os três últimos (lingüístico, representativo e assimilativo), proporcionam a transferência, o entendimento e a aprendizagem do conhecimento que possibilitam utilizá-lo.

## 2.3 Definições de Gestão do Conhecimento

A Gestão do Conhecimento possui definições variadas, de acordo com autores diferentes [NONAKA, 1991; DAVENPORT e PRUSAK, 1998; ZACK, 1999; LAWTON, 2001; SIVAN, 2001; GUPTA et. al., 2002], porém, de maneira geral, o objetivo final é semelhante, ou

seja, gerenciar conhecimentos existentes e adquirir novos conhecimentos, vislumbrando a melhoria de determinado processo, atividade ou organização.

Para Gupta et. al. (2002), Gestão do Conhecimento é basicamente o gerenciamento do conhecimento e do patrimônio intelectual de uma empresa ou organização, que podem contribuir para a melhoria da performance organizacional e agregar valor à organização, permitindo que esta realize suas tarefas de forma mais inteligente e eficiente.

Este mesmo autor complementa que Gestão do Conhecimento é um processo que ajuda as organizações a identificar, selecionar, organizar, disseminar e transferir informações e habilidades que são parte da memória organizacional existente - de forma desestruturada - dentro de uma organização. O processo de Gestão do Conhecimento possibilita a resolução de problemas de forma eficiente e efetiva, facilita o aprendizado dinâmico, o planejamento estratégico e o processo de tomada de decisão. O foco da Gestão do Conhecimento é identificar conhecimento e explicitá-lo de forma que o mesmo possa ser compartilhado formalmente e reutilizado.

Para Sivan (2001), Gestão do Conhecimento é a arte de realizar ações de conhecimento tais, como, organizar, filtrar, compartilhar e disseminar, utilizando objetos do conhecimento como informações, experiências, avaliações, *insights*, sabedoria e iniciativas.

Já Lawton (2001) diz que a Gestão do Conhecimento tem como objetivo capturar conhecimento documentado ou pessoal e outros tipos de conhecimento e torná-los disponíveis de forma que ajudem organizações a atingirem seus objetivos.

Hussain et. al. (2004) pontua que a Gestão do Conhecimento possibilita a transferência de conhecimento de uma pessoa para outra, a qual pode, então, utilizar o conhecimento adquirido. O conhecimento pode ser “alavancado” dentro de uma organização através de iniciativas como:

- compartilhamento de conhecimentos e boas práticas;
- incentivo a cultura de compartilhar o conhecimento;
- introdução de conhecimento em produtos, serviços e processos;
- geração de conhecimento como um produto;
- direcionamento da geração de conhecimento para a inovação;
- mapeamento de redes de especialistas;

- entendimento e avaliação do valor do conhecimento;
- estímulo ao patrimônio intelectual.

Com base nas definições apresentadas acima, pode-se dizer que o processo de Gestão do Conhecimento visa manter, de forma organizada e acessível a qualquer indivíduo, informações que são úteis e, muitas vezes, indispensáveis, para o sucesso de determinada atividade.

## **2.4 Vantagens da Gestão do Conhecimento**

Em um mundo cada vez mais competitivo, a Gestão do Conhecimento torna-se uma aliada que permite controlar e otimizar processos, e criar alternativas inovadoras, mais eficientes e focadas. Isso pode se obtido a partir da identificação do conhecimento necessário, da localização deste conhecimento e de seu compartilhamento e aperfeiçoamento.

Uma organização que é capaz de aprender, segundo Garvin (2000), dispõe de habilidades para criar, adquirir e transferir conhecimento, e consegue modificar seu comportamento de modo a refletir estes novos conhecimentos e idéias. Estas organizações são habilidosas em atividades como: solução de problemas de maneira sistemática, experimentação de novas abordagens, aprendizado com as próprias experiências, aprendizado com as experiências e melhores práticas alheias, e transferência de conhecimento de forma rápida e eficiente.

O autor acima defende que a Gestão do Conhecimento possibilita evoluir do conhecimento superficial para um conhecimento mais profundo, que distingue o saber “como fazer” do “por que fazer”. Saber “como fazer” é conhecimento parcial que se baseia em normas de comportamento e em padrões de prática. Saber “por que fazer” é mais fundamental, pois captura as relações de causas e efeitos, admite exceções, adaptações e eventos inesperados.

Para Mathi (2004), o conhecimento agrega maior valor a produtos, pessoas e processos. O conhecimento aplicado em tecnologia implica em ferramentas e infra-estrutura para a criação de produtos inteligentes que tragam maiores benefícios para seus usuários. Em se tratando de

peças, o conhecimento é o maior patrimônio e, se for bem gerido, proporcionará vantagem sobre os concorrentes e uma maior produtividade.

Segundo Bhirud (2005), a Gestão do Conhecimento é uma ferramenta para criar conhecimento. Através da gerência do conhecimento existente e do compartilhamento deste conhecimento é que surge a inovação.

Organizações podem obter vários benefícios através da implantação de Gestão do Conhecimento, como a redução na perda de capital intelectual quando pessoas trocam de projeto ou saem da empresa, redução na redundância de conhecimento e aumento da produtividade tornando o conhecimento disponível rapidamente e facilmente [HUSSAIN et. al., 2004].

## 2.5 Fatores Chave da Gestão do Conhecimento

Os fatores chave para o sucesso de um projeto de Gestão do Conhecimento podem variar de acordo com as características do projeto. Para Hussain et. al. (2004), a gestão efetiva do conhecimento acontece com base em vários fatores, porém com especial atenção a três: cultura, estratégia e tecnologia.

Com relação ao fator **cultura**, o autor destaca que a Gestão do Conhecimento está intimamente relacionada ao fator humano, ou seja, um projeto deste tipo não terá sucesso caso não seja desenvolvida uma cultura que enfatize o papel e o valor do conhecimento em atividades do dia-a-dia e em processos de tomada de decisão. A cultura deve ser direcionada para a inovação, aprendizagem, experimentação e reflexão.

A regra, neste caso, deve ser voltada para a criação, transferência e uso do conhecimento. De maneira geral, se a cultura de uma organização ou dos participantes de um projeto não estiver intimamente ligada ao processo de Gestão do Conhecimento, não existe tecnologia, repositório de conhecimento ou boas práticas que levem ao sucesso.

As pessoas devem ser orientadas ao conhecimento, ou seja, devem ter curiosidade, força de vontade e liberdade para explorar novas possibilidades, não devem ter receio de compartilhar o conhecimento e devem ter total apoio neste tipo de atitude.

Outro ponto importante para a infra-estrutura de um projeto de Gestão do Conhecimento, de acordo com o autor, é arquitetar, implementar e integrar uma **estratégia** efetiva para implantação do projeto. A grande questão é a criação de um ambiente para disseminar o conhecimento de maneira colaborativa, tornando-o concreto.

Hussain et. al. (2004) afirma que, de maneira geral, a implementação de uma metodologia de Gestão do Conhecimento segue sete passos:

1. Identificação do problema: muitas vezes o conhecimento existe, porém de forma isolada. É necessário identificar as ilhas de conhecimento e os segmentos que as separam.
2. Preparação para mudanças: mudanças são necessárias para alinhar os objetivos e obter sucesso.
3. Definição de uma equipe: em muitos casos uma equipe é definida para a implantação de um projeto piloto.
4. Mapeamento do conhecimento: identificar o que é o conhecimento, onde está, quem o possui e quem precisa dele. Após o mapeamento, devem ser identificadas tecnologias que podem ser usadas para implementar um sistema de Gestão do Conhecimento.
5. Criação de um mecanismo de *feedback*: um sistema de *feedback* deve ser criado para avaliar o processo a fim de identificar eventuais dificuldades.
6. Definição de componentes: um sistema de Gestão do Conhecimento deve ser composto por um repositório de conhecimento, um sistema de acesso ao conhecimento e um gerenciador de conteúdo.
7. Integração com sistemas de informação existentes: a integração com sistemas existentes tem o objetivo de contribuir para a disseminação e captura do conhecimento.

O terceiro fator chave para a Gestão do Conhecimento, segundo o autor, é a **tecnologia**. A tecnologia é uma grande aliada no processo de aquisição e compartilhamento de conhecimento, sendo um bom exemplo de ferramenta de Gestão do Conhecimento.

Um exemplo de uso de sistemas computadorizados em benefício do conhecimento, são sistemas que armazenam conhecimento obtido de especialistas em determinada área, na forma de fatos, símbolos e regras, e disponibiliza este conhecimento para que outras pessoas menos experientes possam aprender. Todo este processo envolve quatro fases:

1. Obtenção do conhecimento de especialistas ou de outras fontes.
2. Representação do conhecimento através de fatos, símbolos e regras.
3. Inferência a partir do conhecimento, ou seja, formulação de conclusões.
4. Transferência de conhecimento para os usuários.

## **2.6 Projetos de Gestão do Conhecimento**

Conforme mencionado anteriormente, as características e as necessidades de um projeto de Gestão do Conhecimento estão diretamente relacionadas ao ambiente no qual o mesmo será desenvolvido. Em outras palavras, isto significa dizer que o processo de desenvolvimento e implantação de projetos de Gestão do Conhecimento podem ser bastante diferentes, apesar de o objetivo final ser semelhante.

Ao mesmo tempo, projetos diferentes podem apresentar fatores semelhantes que podem servir para indicar características que são desejáveis na maior parte dos projetos e que podem determinar seu sucesso.

Apesar das diferenças presentes em projetos de Gestão do Conhecimento, é possível identificar-se algumas características comuns entre os mesmos, as quais podem ser indicadas como fatores que levam ao sucesso. Estas características são descritas ao longo desta seção.

### **2.6.1 Cultura Orientada para o Conhecimento**

Segundo Davenport e Prusak (1998), uma cultura voltada para o conhecimento é uma das condições mais importantes para o sucesso de um projeto. Esta cultura é fruto de uma série de componentes diferentes, como:

- Orientação positiva para o conhecimento: os funcionários têm sede de conhecimento, procuram explorar e criar, e têm liberdade para isso.

- Ausência de inibidores do conhecimento na cultura: não existem ressentimentos em relação à empresa por parte dos colaboradores e os mesmos não temem que o compartilhamento do conhecimento possa lhes custar o emprego.
- O projeto de Gestão do Conhecimento é compatível com a cultura.

Em uma cultura orientada ao conhecimento, o constante aprendizado, dentro ou fora do ambiente de trabalho, é altamente valorizado, juntamente com a experiência e a capacidade de inovação.

Em um ambiente de aprendizagem é permitido errar e aprender a partir dos erros, as experiências não são escondidas, mas, ao contrário, são abertas a quem tem interesse e necessidade. As experiências não são utilizadas para substituir, degradar ou avaliar alguém, mas sim para ajudar. As pessoas são encorajadas a compartilhar experiências e ajudar umas as outras e, além disso, são recompensadas de acordo com o quanto compartilham [RUS et. al., 2001].

### **2.6.2 Infra-estrutura Técnica e Organizacional**

Para Davenport e Prusak (1998) e Mathi (2004), projetos de Gestão do Conhecimento têm maior probabilidade de sucesso quando possuem uma infra-estrutura ampla de tecnologia e de organização.

A tecnologia, segundo Davenport e Prusak (1998), é mais fácil de implementar consistindo basicamente de tecnologias orientadas ao conhecimento, como por exemplo, o Lotus Notes<sup>1</sup> e a Internet. Outra consideração com relação à tecnologia é a necessidade de existência de um conjunto uniforme de tecnologias para computação e comunicação, o que facilita o intercâmbio de informações entre as pessoas.

Rus et. al. (2001) complementa que todos os sistemas de gerenciamento de conhecimento devem estar interligados com outros sistemas de informação, observando-se é claro, questões de segurança e de proteção de informações.

---

<sup>1</sup> <http://www-142.ibm.com/software/sw-lotus/products/product4.nsf/wdocs/noteshomepage>

Já a infra-estrutura organizacional para a Gestão do Conhecimento é algo mais difícil. A mesma consiste em estabelecer um conjunto de regras e novas funções, além de estruturas organizacionais e qualificações que beneficiam cada projeto [DAVENPORT e PRUSAK, 1998].

### **2.6.3 Apoio Gerencial**

Segundo Davenport e Prusak (1998), o apoio por parte da gerência de uma empresa ou organização qualquer é de grande importância para projetos de Gestão do Conhecimento, principalmente quando o projeto envolve transformações do conhecimento. Alguns exemplos de apoio são:

- mensagens à organização ressaltando que a Gestão do Conhecimento e a aprendizagem organizacional são fundamentais para o sucesso da empresa;
- apoio a investimentos na infra-estrutura para a Gestão do Conhecimento;
- esclarecimento com relação ao tipo de conhecimento que é mais importante dentro da organização.

Para os autores, geralmente as pessoas participantes da gerência que mais defendem os projetos de Gestão do Conhecimento, são pessoas mais conceituais e orientadas para o conhecimento.

### **2.6.4 Vinculação ao Valor Econômico e Setorial**

Um projeto de Gestão do Conhecimento requer investimento e se espera, portanto, que o mesmo gere algum benefício econômico ou a conquista de sucesso no setor. Nem sempre é simples quantificar os benefícios provenientes de um projeto deste tipo, porém é importante ter uma maneira de identificar benefícios mesmo que indiretos para justificar o investimento no desenvolvimento do projeto [DAVENPORT e PRUSAK, 1998].

### **2.6.5 Clareza de Visão e Linguagem**

Em um projeto de Gestão do Conhecimento é importante que haja clareza de propósito e objetividade, além de uma terminologia comum dentro do projeto. De uma maneira geral, os principais termos utilizados em projetos de Gestão do Conhecimento – informação, conhecimento e aprendizado – estão sujeitos a um grande número de interpretações diferentes [DAVENPORT e PRUSAK, 1998].

Segundo Rus et. al. (2001), manter a clareza e ter consciência da missão da organização e de sua direção é de extrema importância para que o projeto não perca foco e não seja prejudicado pelo fato de nem todos os seus integrantes estarem alinhados com os objetivos a serem alcançados.

### **2.6.6 Elementos Motivadores**

O conhecimento está diretamente relacionado ao ego e à ocupação das pessoas e, por isto, não emerge nem flui facilmente. Assim, é necessário motivar as pessoas a criar, compartilhar e usar o conhecimento. Os fatores motivacionais não podem ser simples, tem que realmente cativar as pessoas e encorajá-las a participar do projeto e a oferecer seu conhecimento como contribuição [DAVENPORT e PRUSAK, 1998].

Em geral, formas mais eficientes de motivação para encorajar comportamentos relacionados ao conhecimento são os incentivos de longo prazo que devem estar ligados à estrutura de avaliação e remuneração.

Segundo Rus et. al. (2001) as pessoas precisam estar motivadas para compartilhar seu conhecimento com outras pessoas. Estas pessoas precisam estar convencidas de que o compartilhamento de sua sabedoria tem valor para a organização a qual pertencem e que isto é importante para eles próprios.

### **2.6.7 Estruturação do Conhecimento**

Para Davenport e Prusak (1998), um fator crítico, em um projeto de Gestão do Conhecimento, é alcançar um nível ideal de estruturação do conhecimento, pois como o conhecimento é estreitamente ligado às pessoas que o detém, suas categorias e significados mudam freqüentemente.

Se um repositório de conhecimento for totalmente desestruturado, provavelmente será muito difícil se obter o conhecimento que está armazenado nele. Torna-se necessário criar categorias e alguns termos principais, ou até mesmo um *thesaurus* para auxiliar as pessoas que irão acessar esta fonte de conhecimento.

Outro fator interessante é que a estrutura do conhecimento continua em evolução constante e por isto é importante ter alguém responsável por manter esta estrutura atualizada e refletindo o padrão de uso [RUS et. al., 2001].

### **2.6.8 Múltiplos Canais para Transferência do Conhecimento**

O conhecimento é transferido através de diferentes canais que acabam se reforçando mutuamente. A transferência de conhecimento através de vários canais adiciona valor de maneiras diferentes ao projeto e a sinergia entre os canais incentiva o uso do conhecimento [RUS et. al., 2001].

Um exemplo, dado por Davenport e Prusak (1998), é o de algumas empresas que possuíam repositórios de conhecimento ou outros sistemas voltados para o conhecimento, mas que perceberam que além destes recursos era necessário reunir as pessoas em um ambiente que propiciasse um contato cara a cara. Isto permite o estabelecimento de relações de confiança, o desenvolvimento das estruturas de conhecimento e a resolução de problemas difíceis.

### **2.6.9 Avaliação**

A avaliação de um projeto de Gestão do Conhecimento é essencial para que o mesmo mantenha-se alinhado com o ambiente em que está sendo aplicado, para determinar sua eficiência e, além disso, para a melhoria constante do projeto. Estas medidas permitem aumentar o valor agregado do projeto e/ou prolongar a duração da vantagem competitiva obtida com o mesmo [RUS et. al., 2001; MATHI, 2004].

## **2.7 Gestão do Conhecimento e Desenvolvimento de Software**

A Gestão do Conhecimento é uma técnica que pode ser aplicada aos mais variados tipos de atividades, sendo que uma das atividades na qual esta técnica pode ser muito útil é o desenvolvimento de software.

Desenvolver software é uma atividade que está intimamente ligada ao conhecimento, ou seja, o processo de desenvolvimento consiste na aplicação de conhecimentos tácitos e explícitos que podem variar entre uma série de conhecimentos técnicos, como linguagens de programação e padrões de projeto, até o conhecimento das regras de negócio da aplicação que será desenvolvida.

### **2.7.1 Capital Intelectual**

Uma característica especial do desenvolvimento de software é que o ativo mais valioso da organização não são máquinas, equipamentos e instalações, mas sim o capital intelectual das pessoas que participam do processo de desenvolvimento [WALZ, 1993; DINGSOYR e ROYRVIK, 2003; SOARES, 2005].

Dentro do contexto de desenvolvimento de software cada projeto é único, caracterizado por interações com o meio externo e fortemente baseado no conhecimento [SOARES, 2005].

Em projetos desta natureza é necessário que exista uma diversidade de pessoas que possuam conhecimentos variados para que todas as fases do projeto sejam finalizadas.

O desenvolvimento de software é um tumultuado processo, que engloba tecnologias computacionais que evoluem rapidamente e, além disso, é composto por equipes de profissionais altamente qualificados, os quais necessitam utilizar a criatividade e a inovação no processo de desenvolvimento. Se o processo não for bem gerenciado, ele pode virar um caos e acabar comprometendo as datas de entrega do software e também seu orçamento [BASKERVILLE e PRIES-HEJE, 1999].

Para Desouza (2003), o desenvolvimento de software é uma atividade fortemente baseada em conhecimento, na qual o sucesso está intimamente relacionado às experiências das pessoas que participam do desenvolvimento em uma ou mais das seguintes áreas: projeto de sistemas, codificação, teste e implantação. Segundo o autor, várias empresas de tecnologia da informação desenvolveram sistemas de gerência de conhecimento para que seus colaboradores pudessem explorar seus conhecimentos e aprender a partir das experiências de outras pessoas.

## **2.7.2 Compartilhamento do Conhecimento**

De acordo com Walz (1993), mais da metade do custo de desenvolvimento de um sistema de informação complexo está relacionado a decisões tomadas na parte inicial do processo de desenvolvimento, ou seja, na especificação dos requisitos e no projeto do sistema. O compartilhamento do conhecimento adquirido por outras equipes na definição de requisitos e nas decisões de projeto proporciona descobertas que melhoram a qualidade e a produtividade do desenvolvimento.

O autor pontua que o conhecimento é a matéria-prima das equipes de desenvolvimento de software e que, através da aquisição, compartilhamento e integração do conhecimento, os membros da equipe poderão aprimorar o processo de elaboração de um sistema de informação e, conseqüentemente, alcançar resultados melhores ao final do projeto, além de obter conhecimento que será útil nos próximos projetos.

Para Natali e Falbo (2002), a Gestão do Conhecimento pode ser utilizada no contexto de desenvolvimento de software para capturar o conhecimento e a experiência gerados durante o processo de desenvolvimento. Apesar de cada projeto ser único, o compartilhamento de experiências similares pode ser útil para que os desenvolvedores realizem suas tarefas com maior eficácia. A reutilização do conhecimento pode prevenir a repetição de erros passados e guiar a solução de problemas recorrentes.

Estes autores afirmam que lições aprendidas durante o processo de desenvolvimento de software são um dos mais importantes tipos de conhecimento (informal) que podem ser gerados neste ambiente, pois são provenientes do trabalho da própria organização. Tanto as lições aprendidas a partir do sucesso quanto as aprendidas a partir do fracasso devem ser aproveitadas e compartilhadas. O reuso deste conhecimento promove boas práticas de desenvolvimento e previne a repetição de problemas.

### 2.7.3 Aquisição do Conhecimento

De acordo com o que foi mencionado anteriormente, o desenvolvimento de software é uma atividade que utiliza intensamente o conhecimento e desta forma pode tirar proveito das técnicas de Gestão do Conhecimento para aprimorar seus resultados.

Segundo Birk et. al. (1999), o desenvolvimento de software necessita de uma série de atividades baseadas em conhecimento, como a análise de requisitos para novos sistemas, identificação e aplicação das melhores práticas de desenvolvimento, coleta de informações sobre planejamento de projetos e gerência de riscos entre outras.

Rus et. al. (2001) identificou três categorias principais de ações dentro de projetos de desenvolvimento de software em que o conhecimento está fortemente presente e pode ser capturado e formalizado:

- **Tarefas realizadas com o objetivo de desenvolver um sistema baseado em requisitos do cliente:** para os autores esta é uma das principais tarefas. Nesta fase, o gerente do projeto precisa garantir que o projeto será entregue no prazo definido, com o orçamento estipulado, com as funcionalidades requisitadas e com a qualidade esperada. Os documentos criados durante o desenvolvimento do projeto, como

contratos, plano de projeto, especificações de requisitos, códigos fonte e planos de teste, entre outros, possuem informações importantes que, durante o desenvolvimento, documentam as decisões e, após a finalização do projeto, representam sua história e podem então ser utilizados como fonte de aprendizado para os novos projetos.

- **Tarefas que visam melhorar a habilidade dos desenvolvedores:** os autores pontuam que estas tarefas podem ser realizadas durante o projeto ou logo após a finalização do mesmo e têm como objetivo garantir que o conhecimento em potencial obtido durante o projeto não seja perdido. As tarefas consistem em analisar dados do projeto com a intenção de criar conhecimento e poder armazenar este conhecimento em repositório e bases de conhecimento.
- **Tarefas que visam melhorar a habilidade da organização no desenvolvimento de software:** estas tarefas buscam analisar os resultados de projetos anteriores e identificar similaridades e diferenças entre eles. A partir desta análise podem ser definidos padrões e melhores práticas com base em diferentes fontes. Este exercício de aprendizagem possibilita que a instituição evolua a partir de suas próprias experiências e obtenha melhores resultados nos próximos projetos.

A partir destas observações é possível identificar maneiras de se obter um conhecimento valioso que possibilita a melhoria de desempenho das pessoas que participam dos projetos e da organização como um todo. A existência de iniciativas de Gestão do Conhecimento neste ambiente propicia o melhor aproveitamento dos recursos disponíveis dentro da organização.

### 3 DESIGN RATIONALE

O termo *rationale*, que vem do inglês, significa as razões ou intenções de um conjunto de pensamentos ou ações, ou seja, o que motivou estes pensamentos e ações<sup>2</sup>.

Diferente do processo padrão de documentação, que consiste na descrição do resultado final de um projeto, a utilização de técnicas de *Design Rationale* busca documentar, além da decisão final, também as razões por trás de cada decisão, incluindo as justificativas, as alternativas consideradas e os argumentos que levaram a determinada decisão [MORAN e CARROLL, 1996; LEE, 1997].

Segundo Souza et. al. (1999), o registro de informações relativas à decisão é muito importante, pois soluções discutidas e adotadas em um projeto podem ser relevantes também para outros projetos, uma vez que erros já cometidos poderiam ser evitados e alternativas já investigadas poderiam ser melhor aproveitadas. Neste contexto, de acordo com estes autores, *Design Rationale* é o conjunto de informações relacionadas não somente às decisões, mas também relativo às razões que direcionaram cada decisão, incluindo as alternativas consideradas, as justificativas e os argumentos que conduziram à decisão.

A utilização de técnicas de *Design Rationale* em determinado projeto permite que, no momento em que este projeto for revisado por pessoas diferentes, seja possível identificar os motivos pelos quais as decisões foram tomadas, bem como quais eram as alternativas consideradas e porque estas foram rejeitadas, evitando a repetição de trabalho já feito anteriormente.

Para Horner e Atwood (2006), quando o *Design Rationale* é capturado e estruturado, o mesmo pode ser utilizado por pessoas que estão por fora do contexto do projeto para analisar artefatos e a influência das decisões tomadas durante o desenvolvimento do projeto.

Segundo Preece et. al. (1994), já há algum tempo a documentação durante o processo de desenvolvimento de software tem sido uma preocupação da engenharia de software. Muitas pessoas envolvidas neste processo precisam saber as razões por trás das decisões do projeto, de forma que estas informações possam ser úteis tanto para reuso futuro quanto para atividades de manutenção do sistema.

---

<sup>2</sup> <http://dictionary.cambridge.org/>

É importante observar que quando se fala em documentar as decisões do projeto, está se referindo as decisões-chave tomadas durante o projeto, pois, se todas as decisões forem documentadas em detalhe, logo a documentação irá se tornar muito extensa e sua utilidade será prejudicada pela dificuldade de se extrair informações que realmente interessam.

De acordo com Shipman e McCall (1997), existem três perspectivas diferentes de *Design Rationale*: argumentação, comunicação e documentação. No caso da argumentação e da documentação, a captura de informações acontece de forma estruturada, facilitando o armazenamento e, conseqüentemente, a recuperação das informações armazenadas. Já na comunicação, as informações não são capturadas de forma ordenada, o que dificulta a recuperação das mesmas posteriormente.

### 3.1 Abordagens de *Design Rationale*

Dentro do contexto de *Design Rationale*, são encontradas algumas abordagens desenvolvidas para tentar resolver o problema de documentar as decisões de projeto de modo que estas se tornassem acessíveis e reutilizáveis. A seguir, são apresentadas as principais abordagens de *Design Rationale* conhecidas.

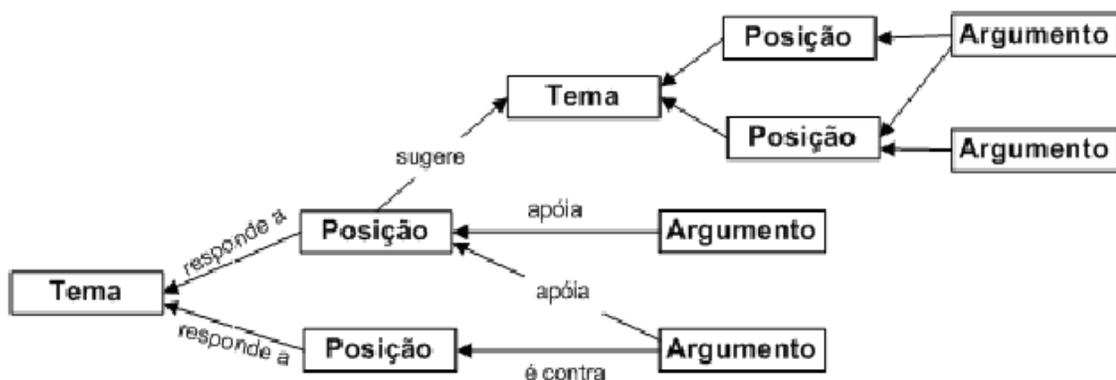
#### 3.1.1 IBIS

O IBIS (*Issue-Based Information System*), de acordo com Kunz e Rittel (1970), é uma abordagem que foi desenvolvida para capturar decisões de projeto durante o progresso do projeto, ou seja, a idéia é que o IBIS seja um produto do processo de projeto de um sistema. Sua idéia principal é considerar os prós e contras das possíveis respostas às questões de projeto. As questões são chamadas de **temas**, as respostas de **posições** e os prós e contras de **argumentos** [PREECE et. al.,1994].

Uma discussão em torno do IBIS começa com o tema principal, que é a principal questão a ser respondida. Posições são obtidas para esta questão juntamente com argumentos

a favor e contra as posições. Temas secundários, terciários, etc., podem ser gerados e tratados da mesma maneira, até que uma solução seja encontrada.

Após a solução ter sido encontrada, os temas são colocados em um diagrama chamado de mapa de temas, que representa suas relações, como pode ser visto na figura 3.1. Além das relações presentes na figura 3.1, várias outras relações entre temas são reconhecidas na técnica original do IBIS como, por exemplo, 'mais geral que', 'similar a', 'substituto', 'sucessor temporal de' e 'sucessor lógico de'.



**Figura 3.1:** Exemplo genérico do mapa de temas [MEDEIROS, 2006].

Algumas outras abordagens foram desenvolvidas com base no IBIS, como por exemplo, o PHI (*Procedural Hierarchy of Issues*) [MCCALL, 1991], que tem como objetivo resolver os problemas apresentados acima. Segundo Preece et. al. (1994), o PHI difere do IBIS em 4 maneiras:

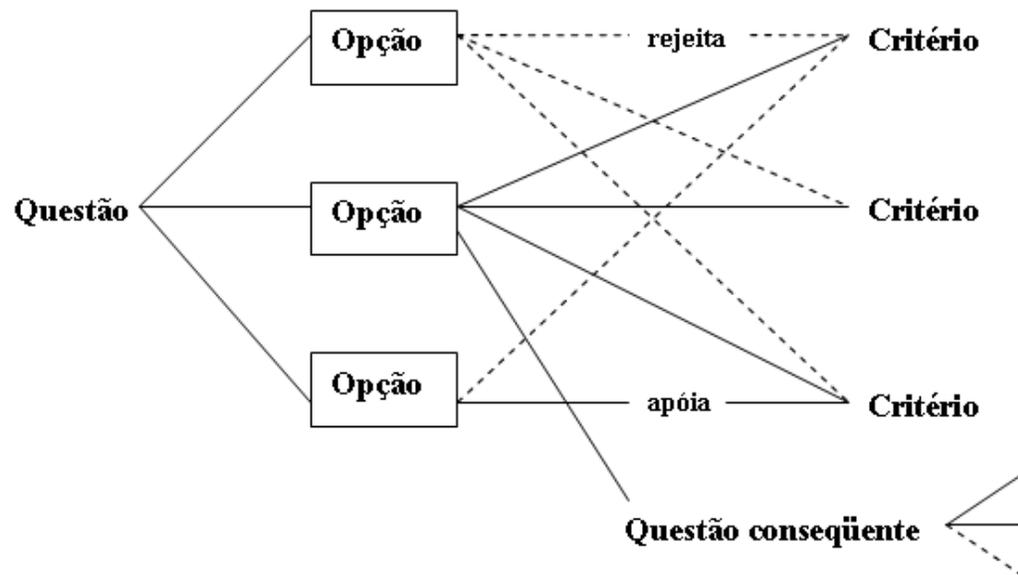
- **O que constitui um tema:** No IBIS original, um tema é considerado uma questão que é deliberada. Ou seja, decisões de projeto que são tomadas sem haver discordância não são consideradas e não fazem parte do mapa de temas. Já no caso do PHI, todas as questões de projeto são consideradas, tendo sido deliberada ou não, e são documentadas como tal.
- **Como relacionar temas:** o PHI introduz um elemento de hierarquia em suas relações, que se trata somente de duas noções: quais temas servem ao tema atual e a que tema o tema atual serve. Segundo Medeiros (2006), um tema A serve a um tema B se resolver A ajuda a resolver B.

- **Como resolver temas:** semelhante ao IBIS, o PHI usa a deliberação para resolver temas, através da análise dos argumentos a favor e contra cada resposta e, então, toma a decisão de qual é a resposta mais correta. Além disso, o PHI estende esta idéia, introduzindo a noção de generalização/especificação de resposta e, ao invés de categorizar os argumentos como sendo a favor ou contra uma possível resposta, simplesmente os considera como argumentos.
- **Como identificar temas:** o PHI é mais restritivo que o IBIS na maneira como os temas são identificados. A hierarquia de temas é gerada de cima para baixo (*top-down*), iniciando pelo tema principal e percorrendo a hierarquia de sub-temas até que nenhum outro seja identificado.

### 3.1.2 QOC

O QOC (*Questions, Options and Criteria*) é baseado na formulação de questões, as quais são uma importante dimensão do espaço de projeto, representando os principais problemas que devem ser considerados e que compõem o espaço de alternativas. Critérios são objetivos positivos que servem como base para a avaliação e escolha entre as várias opções. E opções são as possíveis respostas às questões [MACLEAN et. al., 1991].

Dentro da representação desta técnica, linhas contínuas representam relações positivas entre um critério e uma opção, enquanto linhas pontilhadas indicam uma relação negativa, ou seja, a opção não é suportada pelo critério. Um exemplo dessa representação é mostrado na figura 3.2.



**Figura 3.2:** Exemplo genérico da notação QOC, adaptado de Shum e Hammond (1994).

Apesar da notação utilizada nesta técnica objetivar a identificação de alternativas de projeto e destacar as opções e critérios a serem usados e a relação entre eles, ela não identifica a melhor opção por conta própria. A decisão de qual deve ser a melhor opção não é simples e vai depender da habilidade do projetista.

Segundo Preece et. al. (1994), os projetistas são altamente encorajados a explorar diferentes alternativas de projeto, pois, assim, o resultado final não é somente uma descrição dos porquês das decisões, mas, também, um resultado de melhor qualidade a medida que o projetista terá que explorar diferentes alternativas.

Outro ponto importante a ser observado é que a documentação de decisões de projeto demanda tempo, o que, em geral, explica por que isso não acontece na maioria dos projetos. Porém, esta técnica abrange importantes problemas de projeto, sendo que, com o surgimento de ferramentas adequadas, talvez seja possível torná-la mais utilizável. Isso pode fazer com que, além de suportar a análise de diferentes alternativas de projeto, também seja possível a reutilização de projetos, assim como acontece algumas vezes no caso de código e, por consequência, isso possibilite que projetistas inexperientes aprendam com outros mais experientes [PREECE et. al., 1994].

De acordo com McKerlie e MacLean (1993), a partir do momento em que as pessoas se acostumam com o QOC elas tendem a usá-lo naturalmente em atividades do dia-a-dia como, por exemplo, durante reuniões sobre um projeto. Os autores complementam que o QOC pode servir como base para documentar as razões envolvidas nas decisões de um projeto, além de permitir a organização das idéias discutidas e até mesmo a organização de outros recursos de representação utilizados em um projeto.

### 3.1.3 DRL

A DRL (*Decision Representation Language*) foi inicialmente desenvolvida como uma linguagem para representar o processo de tomada de decisão, sendo que, posteriormente, foi estendida para a representação de *design rationale* [LEE e LAI, 1991]. Os autores argumentam que uma representação deve suportar um certo número de tarefas de projeto, tais como, responder a questões sobre o progresso do projeto, as alternativas geradas, as avaliações que levaram a escolha de uma alternativa particular e a possível transferência de conhecimento para projetos futuros e outras pessoas. A DRL tem como objetivo expressar estas questões.

A DRL não inclui deliberações sobre como gerar alternativas de projeto, pois sua ênfase é no gerenciamento de elementos qualitativos da tomada de decisão e gerenciamento de dependências, sendo, desta forma, mais do que um sistema de representação da tomada de decisão [STUMPF, 1998].

Este método esgota a avaliação de alternativas através da referência de objetivos explícitos que capturam os objetivos do processo de projeto, em vez de se concentrar na exploração do espaço de projeto. Este método não beneficia o processo de exploração de um problema mas, sim, o gerenciamento do peso de uma decisão em um dado projeto, de forma correta e consistente [LEE, 1991]. A figura 3.3 exhibe os elementos e relações que formam o vocabulário da linguagem DRL.

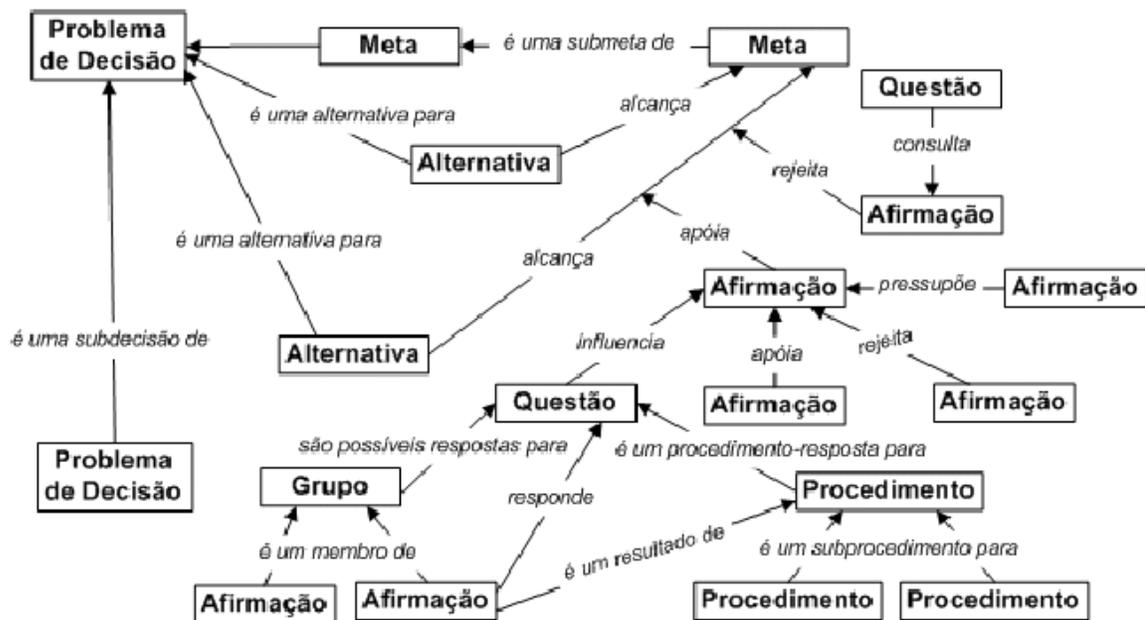


Figura 3.3: Vocabulário da linguagem DRL [MEDEIROS, 2006].

## 3.2 Requisitos para a Representação de *Design Rationale*

Com base nas características e nos principais problemas identificados nas abordagens mais conhecidas de *Design Rationale*, Medeiros (2006), gerou a seguinte lista de requisitos para a representação de *design rationale*:

- 1. Representação explícita de decisões:** Uma mesma alternativa de projeto pode ser considerada, ou não, uma solução para questões de projeto diferentes. Portanto, as decisões tomadas sobre a aceitação ou não desta alternativa como uma solução para cada questão de projeto devem ser registradas separadamente.
- 2. Distinção entre argumentos e justificativa final:** Algumas vezes, o projetista decide aceitar uma idéia de solução que possui apenas argumentos contrários. Podem existir, também, idéias que, pelo número de argumentos a favor, parecem representar uma melhor solução de projeto do que a idéia escolhida pelo projetista para uma dada questão. Isto pode indicar que existe um *rationale* que não está sendo registrado ou que a escolha desta idéia de projeto deve ser reconsiderada pelo projetista. Geralmente o

*rationale* que está faltando pode ser registrado como um novo argumento. Porém, em alguns casos, não se trata de um novo argumento para a idéia de solução proposta, mas, sim, de uma justificativa final para a decisão tomada pelo projetista após analisar os argumentos apresentados. Por exemplo, esta justificativa final poderia registrar os motivos pelos quais o projetista decidiu usar uma idéia de solução, apesar de todos os argumentos apresentados para ela serem contrários à sua aceitação.

**3. Integração da argumentação com as descrições dos artefatos gerados:** A complexidade e o volume de informações presentes nas representações de *design rationale* tornam difícil para o projetista compreender as diversas idéias de solução e o resultado das decisões tomadas. A relação entre a argumentação registrada e o artefato produzido pode facilitar esta compreensão e ajudar o projetista a decidir se vale a pena reusar um artefato ou não.

**4. Registro de informações sobre o contexto de projeto:** Registrar apenas a argumentação usada pelos projetistas não é suficiente para compreender o conhecimento usado por eles durante o projeto. É necessário, também, ter informações sobre o contexto de projeto, como, por exemplo, o método de projeto utilizado, as atividades realizadas e os responsáveis pelas decisões tomadas.

**5. Integração do *design rationale* com o método de projeto:** Representações genéricas e informais de *design rationale* dificultam o uso do *rationale* em novos projetos. A integração do *rationale* com a semântica formal definida pelos métodos de projeto torna o conteúdo registrado mais rico e expressivo, possibilitando a realização de operações computáveis capazes de apoiar novos usos de *design rationale* como, por exemplo, a combinação de *rationales* de artefatos similares.

## 4 ESTUDOS PRELIMINARES

Este capítulo apresenta a caracterização do problema que motivou o desenvolvimento desta dissertação e o escopo deste trabalho, além de um estudo de caso que foi utilizado para embasar a proposta apresentada no capítulo 5.

### 4.1 Caracterização do Problema

Um projeto de software envolve uma grande quantidade de conhecimento, o qual pode estar relacionado à tecnologia utilizada no projeto, à lógica de negócio necessária ao software ou ainda relacionado a outras atividades ou dependências do projeto. Além disso, várias decisões são tomadas durante o desenvolvimento de um sistema e, mesmo após sua conclusão, em atividades de manutenção.

Considerando a grande quantidade de informação existente em projetos de software e o volume de decisões tomadas ao longo do seu desenvolvimento, é importante a existência de alguma ferramenta para documentar e manter estas decisões e, especialmente, os motivos pelos quais estas foram tomadas.

Esse tipo de documentação serve para que as pessoas envolvidas no projeto tenham uma visão mais ampla do mesmo e possam contribuir mais efetivamente, pois terão maior propriedade sobre seus detalhes. Isto é confirmado por Garvin (2000), que afirma que saber “por que fazer” determinada atividade é fundamental, pois captura as relações de causa e efeito, admite exceções, adaptações e eventos inesperados.

Outro fator a se considerar é que esta documentação pode ser útil para pessoas novas no projeto e, até mesmo, servir para reuso no futuro ou em outros projetos, o que segundo Gupta et. al. (2002) é o foco da Gestão do Conhecimento, ou seja, explicitar o conhecimento para que o mesmo possa ser compartilhado e reutilizado.

Dentro deste contexto, a Gestão do Conhecimento propõe a organização e o compartilhamento de todo o conhecimento importante a uma determinada atividade [NONAKA, 1991; DAVENPORT e PRUSAK, 1998; ZACK, 1999; GUPTA et. al., 2000; LAWTON, 2001;

SIVAN, 2001]. De forma semelhante, as abordagens de *Design Rationale* buscam manter, de forma estruturada e organizada, as decisões-chave tomadas durante o desenvolvimento de um projeto, para que estas possam ser úteis especialmente em atividades futuras [MORAN e CARROLL, 1996].

Por estas razões, esta pesquisa visa aliar aspectos do *Design Rationale* à Gestão do Conhecimento e, a partir disto, obter uma representação que possibilite organizar e disponibilizar as principais decisões de projetos de software. A motivação para isso está no fato de que, de acordo com os conceitos de Gestão do Conhecimento e *Design Rationale*, manter as decisões e as respectivas razões oferece maior vantagem competitiva a um projeto e possibilita maior troca de experiência até entre integrantes de diferentes projetos.

Com base nas necessidades e nos benefícios, recém citados, de se ter uma forma para manter as decisões de projeto, partiu-se para a análise de algumas abordagens existentes, com o objetivo de identificar suas forças e fraquezas como método para registro de decisões. Ao mesmo tempo, foram pesquisadas as necessidades de profissionais que trabalham em projetos de software em termos de documentação de decisões para, assim, identificar, dentre as abordagens analisadas as características mais importantes e de maior aplicabilidade em projetos de software.

## **4.2 Escopo do Trabalho**

Este trabalho se atém à proposta de uma abordagem para captura e representação das razões por trás das decisões de um projeto.

É necessário salientar que não estão sendo considerados os impactos da utilização desta proposta em um projeto de software, sendo que a avaliação deste impacto demandaria um estudo voltado para identificação de como os projetos podem ser afetados. E, para este estudo seria necessário acompanhar o processo de desenvolvimento de um (ou mais) projeto(s) para assim mensurar as conseqüências da utilização da proposta.

## 4.3 Estudo de Caso

Para se chegar à proposta apresentada no próximo capítulo, duas abordagens de *Design Rationale* (QOC e DRL), além de uma representação textual foram utilizadas para modelar as decisões tomadas em um projeto de software. A partir dessa modelagem foram realizadas entrevistas com profissionais que atuam neste tipo de projeto de forma a se obter o *feedback* sobre as abordagens utilizadas, bem como identificar algumas necessidades reportadas por estes profissionais.

No decorrer deste capítulo detalha-se o sistema usado como base para as modelagens, as modelagens realizadas com as três abordagens escolhidas e, por último, os resultados obtidos nas entrevistas feitas com profissionais da área.

### 4.3.1 Descrição do Sistema Objeto

O sistema utilizado neste estudo de caso foi desenvolvido por uma empresa que desenvolve soluções para a área de recursos humanos e folha de pagamento. O sistema em questão faz parte de um portal colaborativo e sua função é possibilitar a seleção de benefícios dinamicamente, ou seja, o objetivo do sistema é permitir que cada colaborador de determinada empresa tenha a possibilidade de escolher, dentre uma série de benefícios e coberturas disponíveis, aqueles benefícios que o interessam e satisfazem suas necessidades. Por exemplo, um colaborador de uma empresa deseja como benefício um plano de saúde para sua família, logo ele poderá optar dentre os planos de saúde disponíveis no sistema, aquele que oferece cobertura familiar e cujo custo mensal esteja de acordo com suas possibilidades.

Os benefícios e a variedade destes benefícios são dependentes da empresa que adquire o sistema, pois esta empresa é quem decide o que irá oferecer a seus colaboradores.

Desta forma, a cada ano, durante um período específico, o sistema fica disponível para que o colaborador possa selecionar suas preferências, sendo que, após este período, o sistema estará disponível somente para consulta às escolhas previamente feitas. O sistema permite,

também, que o colaborador cadastre dependentes e escolha os benefícios aos quais estes terão direito.

O sistema funciona como um *wizard*<sup>3</sup> em que o usuário navega por diferentes páginas, selecionando as opções que deseja e, ao final, submete suas escolhas para que sejam registradas. Este sistema é composto por quatro páginas principais: seleção de planos de benefícios, listagem e manutenção de dependentes, revisão e confirmação das seleções e finalmente o resumo das opções selecionadas.

Na primeira página, são exibidos os benefícios disponíveis, contendo o nome da empresa que oferece o plano, as opções de cobertura e os valores mensais de participação do colaborador. É nesta parte que as diversas opções de planos, tais como plano de saúde e plano odontológico, entre outras, podem ser selecionadas. Além de optar pelo plano, o colaborador deve definir o tipo de cobertura: individual, incluindo somente esposa/marido, incluindo somente filhos ou incluindo toda a família.

Na segunda página são exibidos os dependentes cadastrados e os planos disponíveis para cada dependente de acordo com as opções selecionadas na página anterior. Neste momento se escolhe quais dependentes estarão cobertos por quais planos. Além disso, a partir desta página é possível incluir, atualizar e excluir dependentes.

A terceira página contém uma revisão das seleções feitas anteriormente e, a partir desta, o usuário pode optar por alterar suas seleções, voltando para a primeira página, ou então confirmá-las. Após a confirmação, a quarta página é exibida com o resumo das opções escolhidas e um cartão temporário que pode ser impresso e utilizado até que um cartão definitivo seja enviado para o colaborador.

Além disso, como o sistema em questão não foi desenvolvido para a necessidade de uma empresa específica, ou seja, o mesmo produto é vendido para várias empresas diferentes, este oferece a opção de customização dos textos (instruções), *labels* e a visibilidade de determinado campo na página. Esta funcionalidade permite que cada empresa customize parcialmente as páginas do sistema de forma a atender melhor suas necessidades e a facilitar a comunicação com seus colaboradores através de um vocabulário familiar e acessível.

O sistema foi desenvolvido em Java, utilizando plataforma web, sendo acessível por seus usuários através de um portal na Internet. Esta aplicação é integrada com outros sistemas já existentes, de onde provêm as informações sobre o colaborador, seus dependentes,

---

<sup>3</sup> Assistente para realização de tarefas.

benefícios e coberturas disponíveis. A função principal desta aplicação é oferecer uma interface integrada e um ponto de acesso único onde os usuários podem visualizar e atualizar informações e opções relacionadas a todos os benefícios oferecidos pela empresa em que trabalham.

#### **4.3.2 Principais Problemas de Decisão Envolvidos no Projeto**

Durante o levantamento de requisitos deste sistema (baseado nas definições oferecidas pelos analistas de negócio, que têm como objetivo atender as necessidades do mercado alvo) e, também, durante o desenvolvimento do sistema, algumas questões precisaram ser resolvidas, sendo as principais destacadas a seguir:

- Quais funcionalidades devem ser desenvolvidas, considerando o tempo disponível e os requisitos?
- O sistema deve oferecer a possibilidade de manutenção de dependentes?
- As páginas do sistema podem ser customizadas de acordo com as necessidades do cliente?
- Se o sistema somente estará disponível para atualização durante determinado período do ano, como deve ser feito o controle de acesso fora deste período?
- O sistema deve oferecer funcionalidade para impressão das seleções?
- Quais validações devem ser feitas durante a seleção dos planos?

Todas estas questões foram observadas com base nos requisitos do sistema, na infraestrutura disponível para o seu desenvolvimento e no cronograma estabelecido para o lançamento do projeto. Algumas das questões, como controle de acesso e customização de páginas, foram motivadas pela existência de ferramentas auxiliares que realizam estas tarefas, porém foi preciso verificar a possibilidade de integrá-las caso estas atendam aos requisitos ou então desenvolver estas funcionalidades separadamente de forma que ofereçam todos os recursos necessários ao projeto.

### 4.3.3 Modelagem das Decisões

Com base nos problemas de decisão citados, foi feita a modelagem das decisões e opções avaliadas utilizando duas abordagens de *Design Rationale*, o QOC e a DRL, e ainda uma abordagem textual. O resultado destas modelagens será apresentado nas subseções seguintes.

#### 4.3.3.1 QOC

A Figura 4.1 exibe a modelagem das decisões do sistema de seleção de benefícios utilizando QOC. Nesta modelagem foram considerados como questões os principais problemas de decisão citados na seção 4.4.2 e, para cada questão, foram definidas opções de resolução. Além disso, de acordo com o QOC, para cada opção foram estabelecidos critérios favoráveis ou não a determinada opção.

#### 4.3.3.2 DRL

A Figura 4.2 apresenta a modelagem das decisões utilizando a DRL. As mesmas questões utilizadas na modelagem com QOC foram consideradas com a DRL. Como pode ser visto nesta figura, a DRL expressa uma quantidade maior de relações do que o QOC, o que pode causar maior dificuldade para visualizar e entender os relacionamentos.

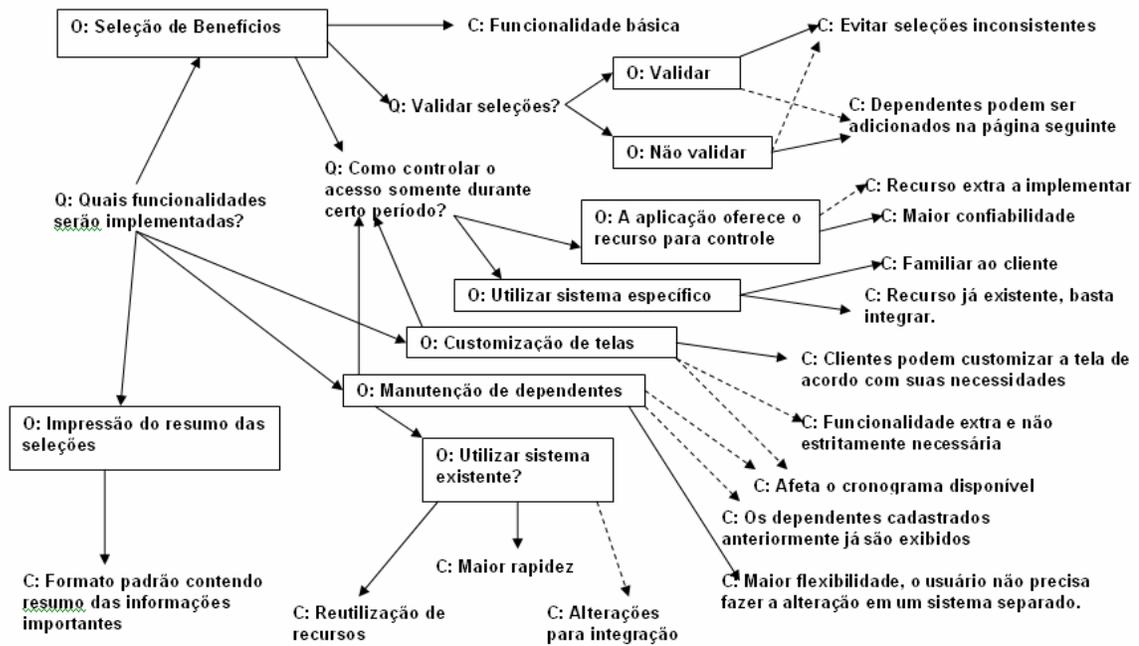


Figura 4.1: Modelagem das decisões utilizando QOC.

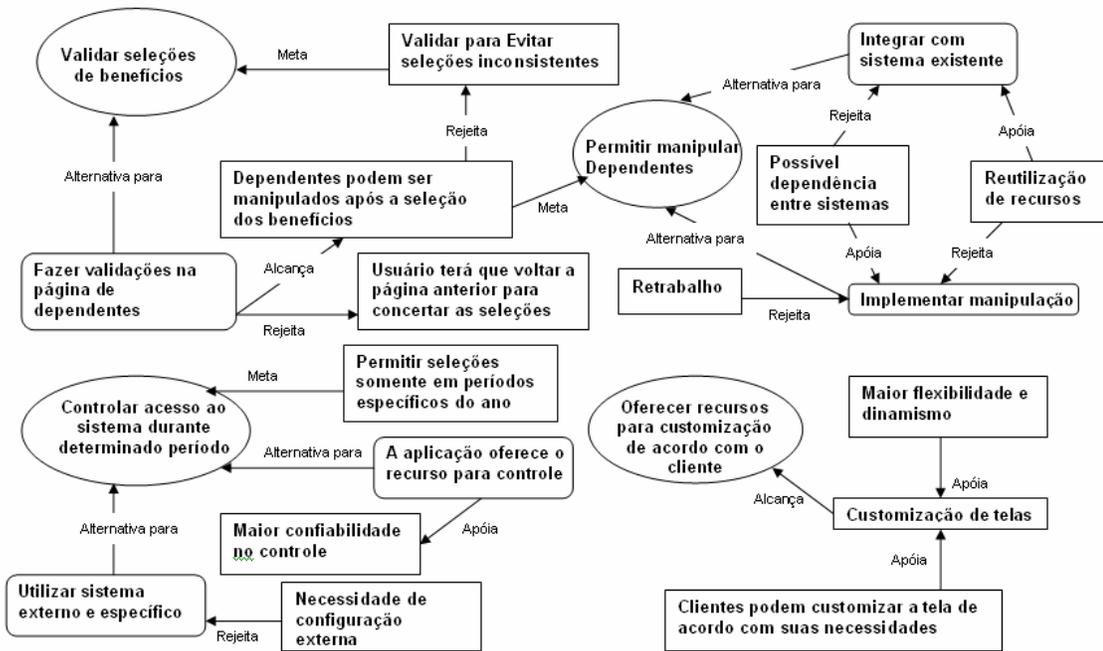


Figura 4.2: Modelagem das decisões utilizando DRL.

#### **4.3.3.3 Modelagem Textual**

A modelagem textual foi utilizada neste projeto como alternativa para avaliar opções para o desenvolvimento do sistema e identificar quais dessas opções melhor se adaptam a realidade e as necessidades do projeto. As decisões tomadas e suas razões são apresentadas na seqüência.

#### ***Funcionalidades a serem implementadas***

Com base no tempo disponível e no orçamento do projeto, optou-se por implementar as seguintes funcionalidades:

- seleção de benefícios para o funcionário e dependentes;
- manutenção dos dependentes (permitindo adição, exclusão e atualização de dependentes);
- customização das páginas de acordo com a necessidade do cliente;
- impressão do resumo das seleções.

As funcionalidades acima foram identificadas como necessárias para o sistema, de forma que o mesmo possa ser lançado no mercado e possa tornar-se competitivo, atendendo as necessidades dos clientes e fazendo frente a sistemas similares existentes no mercado.

#### ***Manutenção de dependentes***

O sistema terá que permitir a manutenção de dependentes, ou seja, o funcionário deve poder adicionar, atualizar ou excluir seus dependentes. Para possibilitar esta funcionalidade, foram levantadas duas hipóteses: a implementação da funcionalidade no sistema ou, então, a integração com outro sistema que já realiza a manutenção de dependentes.

A decisão tomada foi a de integrar o novo sistema ao sistema existente para a manutenção de dependentes. Esta decisão visa reaproveitar uma aplicação pronta ao invés de reimplementar a mesma funcionalidade em outro sistema. Além disso, esta opção facilita a manutenção da aplicação, pois em caso de eventuais alterações estas ocorrerão em um único lugar. Outro fator a se levar em consideração é que os usuários já estão familiarizados com o sistema existente, não havendo impacto de aprendizado.

### ***Controle do acesso ao sistema***

Um dos requisitos do sistema é que o mesmo esteja disponível para seleção de benefícios somente durante determinado período do ano. Ou seja, os usuários somente poderão alterar suas seleções durante este período e, durante o resto do ano, somente será possível visualizar as seleções previamente feitas. Para isso, é necessário um mecanismo que permita controlar o tipo de acesso ao sistema, sendo que duas opções foram consideradas: a lógica de controle de acesso estar na aplicação ou novamente utilizar-se de aplicação externa que realiza este tipo de controle.

Apesar da utilização do sistema externo ser um pouco menos confiável, pois necessita de configuração externa e manual, optou-se por esta alternativa, pois esta já é utilizada largamente em outras aplicações e os usuários estão totalmente familiarizados. Além disso, esta opção oferece alguns recursos avançados que podem ser úteis e oferecem uma maior flexibilidade, como, por exemplo, a possibilidade de configuração de grupos de acesso.

### ***Customização de páginas***

O sistema em questão, assim como outros sistemas desenvolvidos pela empresa são genéricos, ou seja, a mesma aplicação é vendida para vários clientes. Desta forma, surgiu a idéia de possibilitar a customização das aplicações permitindo que os clientes façam pequenas alterações na interface tais como texto de *labels* e instruções, omissão de campos, troca da ordem dos campos na página, mudança das imagens e cores. Apesar do trabalho extra de desenvolvimento, esta funcionalidade é bastante importante, pois permite ao cliente alterar a interface do sistema em certos aspectos e assim adequá-lo a suas necessidades e peculiaridades.

### ***Impressão do resumo das seleções***

Apesar de o sistema ser WEB e já contar com a funcionalidade de impressão dos próprios navegadores, optou-se por oferecer a impressão do resumo das seleções. A motivação para esta funcionalidade é que a impressão do resumo torna-se padronizada e acessível ao usuário a partir de um botão na página.

### **4.3.4 Entrevistas**

Após ter realizado a modelagem do sistema para seleção de benefícios utilizando o QOC, a DRL e uma representação textual, foram realizadas entrevistas com profissionais que trabalham com desenvolvimento de software, para, a partir de suas opiniões, sugestões e experiências em diferentes projetos, obter informações que possam contribuir para o desenvolvimento desta pesquisa.

#### **4.3.4.1 Perfil dos Entrevistados**

As entrevistas foram realizadas com um grupo de 8 profissionais que atuam em projetos de software, sendo 2 gerentes de projeto, 4 líderes de equipe e 2 analistas de sistemas.

Todos os entrevistados possuem ampla vivência na área, com mais de 6 anos de experiência, inclusive com atuação em projetos internacionais. A escolha dos profissionais foi feita com base na função que desempenham dentro do projeto em que atuam, de forma a contar com a opinião de pessoas que participam do processo de tomada de decisões relativas ao projeto.

#### **4.3.4.2 Dinâmica das Entrevistas**

As entrevistas aconteceram através de uma discussão sobre a documentação das razões para as decisões tomadas em um projeto de maneira geral. Em seguida foram apresentadas as três abordagens utilizadas neste estudo: o QOC, a DRL e uma representação textual. Para cada abordagem foi apresentada a representação conceitual da mesma, seguida da modelagem das decisões analisadas para o sistema de seleção de benefícios, descrito anteriormente. Durante a discussão inicial, os entrevistados falaram sobre suas experiências relacionadas à documentação de projetos, sobre as carências que observam em seu dia-a-dia e sobre sua opinião pessoal a respeito do tema.

Ao final, foi discutida a aplicabilidade destes modelos ou de modelos que tenham objetivos semelhantes e foram registradas as observações e sugestões de cada entrevistado. A partir destas informações foi possível compilar uma lista de observações e sugestões a serem analisadas e utilizadas como base para a proposta a ser elaborada.

Apesar do formato das entrevistas ter sido de uma discussão livre, as mesmas seguiram um roteiro padrão, pré-estabelecido, e durante as entrevistas buscou-se as respostas dos entrevistados às questões definidas no roteiro, o qual pode ser visto no Apêndice A.

#### **4.3.4.3 Análise das Entrevistas**

A partir das entrevistas, obteve-se uma série de informações que são provenientes da experiência e das necessidades observadas pelos entrevistados. Estas informações são valiosas, pois retratam a visão de um grupo de profissionais a partir da realidade vivenciada em projetos atuais ou do passado.

Os resultados obtidos com as entrevistas são apresentados na seqüência, sendo que os mesmos estão organizados nas seguintes categorias: observações gerais, observações específicas e sugestões a cada abordagem utilizada. É importante ressaltar que os resultados apresentados a seguir são resultantes de uma compilação por soma das respostas obtidas durante as entrevistas.

##### **4.3.4.3.1 Observações Gerais**

Durante a realização das entrevistas, uma série de observações a respeito do tema discutido foi percebida:

- Todos acreditam que a documentação das decisões-chave de um projeto é útil e importante, pois permite manter o histórico das decisões do projeto.
- Acredita-se que estas informações possam ser utilizadas no próprio projeto em um momento futuro ou, então, como retro-alimentação para utilização em outros projetos que venham a vivenciar situações semelhantes.

- Existe a preocupação do impacto no projeto com a utilização de algumas destas técnicas, principalmente por introduzir uma notação nova e não dominada por todos.
- As modelagens ajudam a observar as dependências existentes no projeto, bem como entender os objetivos e definir melhorias, pois permitem uma visão geral do projeto.
- As modelagens são úteis para questões que anteriormente eram inviáveis, mas por mudanças no projeto passam a ser viáveis e até mesmo necessárias. O fato de se ter o histórico das razões por trás das decisões permite avaliar novamente determinada situação através da comparação do cenário anterior com o novo cenário.
- Há preocupação com a burocratização do processo. É preciso que seja algo simples de se utilizar e fácil de entender (“...processos tem que ser criados e seguidos, mas para serem seguidos não podem ser onerosos...”).
- O fato de se ter as razões documentadas é útil para justificar as decisões tomadas, inclusive para a gerência.
- As questões consideradas como pertinentes para avaliação e documentação foram: decisões relativas às regras de negócio; decisões de arquitetura; decisões que causam grande impacto no cronograma; decisões relativas a ferramentas que serão utilizadas no projeto, e decisões relacionadas a mudanças de requisitos.

#### 4.3.4.3.2 Observações Específicas

Além de observações gerais, foram feitas algumas observações específicas a cada abordagem apresentada. Com relação ao QOC observou-se que:

- É claro, objetivo, simples e de fácil leitura.
- Possui notação clara.
- Proporciona uma visão geral do projeto.
- Alguns entrevistados tiveram dificuldade para encontrar um ponto inicial.

Sobre a DRL foram feitas as seguintes observações:

- É uma representação complexa e confusa.
- É difícil de ler e entender.

- É difícil entender o fluxo e as relações.
- É uma forma de representar, porém pode ser muito subjetiva e confusa.

E sobre a representação textual foram feitas as seguintes observações:

- Proporciona uma descrição rica e detalhada, mas não necessariamente mais clara que as outras representações.
- É subjetiva e passível de interpretação.
- Ninguém gosta de ler muito texto.
- Talvez seja interessante aliar a descrição textual à diagramática, pois a textual possibilita mais detalhes e a diagramática uma visão global.
- É um documento interessante para mostrar para alta gerência, mas não para uso no dia-a-dia.

#### 4.3.4.3.3 Sugestões

Além das observações, algumas sugestões pertinentes foram feitas pelos entrevistados:

- Definir critérios para avaliação dos problemas, pois é necessário saber o que e quando é necessário avaliar.
- Definir e representar o peso dos argumentos, de forma que seja possível identificar as melhores opções. O peso é importante, pois nem sempre o número de argumentos que apóia ou rejeita determinada opção reflete sua validade.
- Permitir segmentar as representações, para que os diagramas não fiquem muito complexos em caso de projetos grandes, ou seja, dividir a representação do projeto em partes, mas sem perder o relacionamento entre elas.
- Aliar a descrição diagramática e textual, proporcionando uma representação esquemática juntamente com um texto explicativo. Permite a visão e o entendimento rápido através do diagrama e ao mesmo tempo oferece descrição detalhada para quem estiver interessado ou precisar da mesma.

## 5 QOC \*

O capítulo anterior apresentou um estudo de caso que contou com a participação de profissionais que atuam e participam das decisões em projetos de software, sendo que, com base neste estudo de caso e no referencial teórico pesquisado, foi desenvolvida a proposta de uma abordagem de *Design Rationale*, denominada QOC \*. Esta abordagem é baseada no QOC e tem como finalidade facilitar o gerenciamento do conhecimento relacionado ao processo de tomada de decisão em projetos de software.

Este capítulo é composto pelo detalhamento da proposta da abordagem QOC \* e pela análise dos resultados de sua utilização por profissionais que atuam em projetos de software e participam diretamente das decisões tomadas nestes projetos.

### 5.1 Características do QOC \*

Através das pesquisas realizadas ao longo deste trabalho, obteve-se uma série de informações pertinentes à documentação de decisões de projeto em geral e às necessidades dos profissionais entrevistados, além de informações e impressões específicas sobre as três abordagens utilizadas. Estas informações foram analisadas com a intenção de identificar as características de cada abordagem que melhor atendessem as necessidades dos profissionais envolvidos no processo de tomada de decisão em projetos de software.

A partir destas informações, chegou-se a conclusão que, das abordagens utilizadas, a que teve maior aceitação pelos profissionais entrevistados foi o QOC, principalmente por possuir uma representação simples e fácil de entender e por instigar a busca de soluções a partir da definição dos principais problemas. A DRL foi considerada uma representação complexa e de difícil leitura, sendo mais suscetível à interpretação do que o QOC. Já a representação textual apresentada, foi considerada interessante por poder ser bastante detalhada, porém é senso comum (e os profissionais também relataram isto) que as pessoas em geral não simpatizam com documentos longos, preferindo representações que sejam mais intuitivas e objetivas.

Com a realização das entrevistas, observaram-se outros aspectos, que não estavam presentes nas abordagens apresentadas para os entrevistados, mas que foram considerados importantes e úteis se agregados a uma abordagem para representar as decisões de um projeto e suas razões. Entre estes aspectos, estão a representação da importância das questões e dos critérios favoráveis ou contrários a determinada opção, pois acredita-se que esta representação ajude a identificar os pontos críticos do projeto, que mereçam maior atenção, e as opções mais apropriadas para atender determinada questão. Outro aspecto considerado é a utilização de uma breve descrição textual dos principais itens da representação, como uma forma de contextualização para quem for utilizá-la posteriormente.

Sendo assim, com base nos resultados da pesquisa, optou-se pela utilização do QOC como ponto de partida para a proposta a ser feita e a partir disso agregou-se características que objetivam tornar a abordagem mais completa e direcionada às necessidades de profissionais da área. As subseções a seguir detalham as características da abordagem que está sendo proposta.

### **5.1.1 Notação Simples**

A notação é baseada no QOC, cuja representação é simples e não possui muitas relações e possibilidades de representação, sendo composta por questões, opções, critérios e relações positivas ou negativas entre as opções e os critérios. Além disso, esta proposta introduz novos itens, como a representação da importância das questões e do peso dos critérios, que serão detalhados nas seções seguintes.

Uma das principais razões pela qual a presente proposta foi desenvolvida com base no QOC é justamente a facilidade de entendimento e representação desta abordagem, confirmada nas entrevistas realizadas. Apesar de não existir uma ferramenta computacional destinada especificamente para este tipo de representação, é relativamente fácil representá-la manualmente ou, então, com o auxílio de algumas ferramentas existentes como o MS Word ® ou o MS Visio ®, entre outras.

### 5.1.2 Representação da Importância das Questões

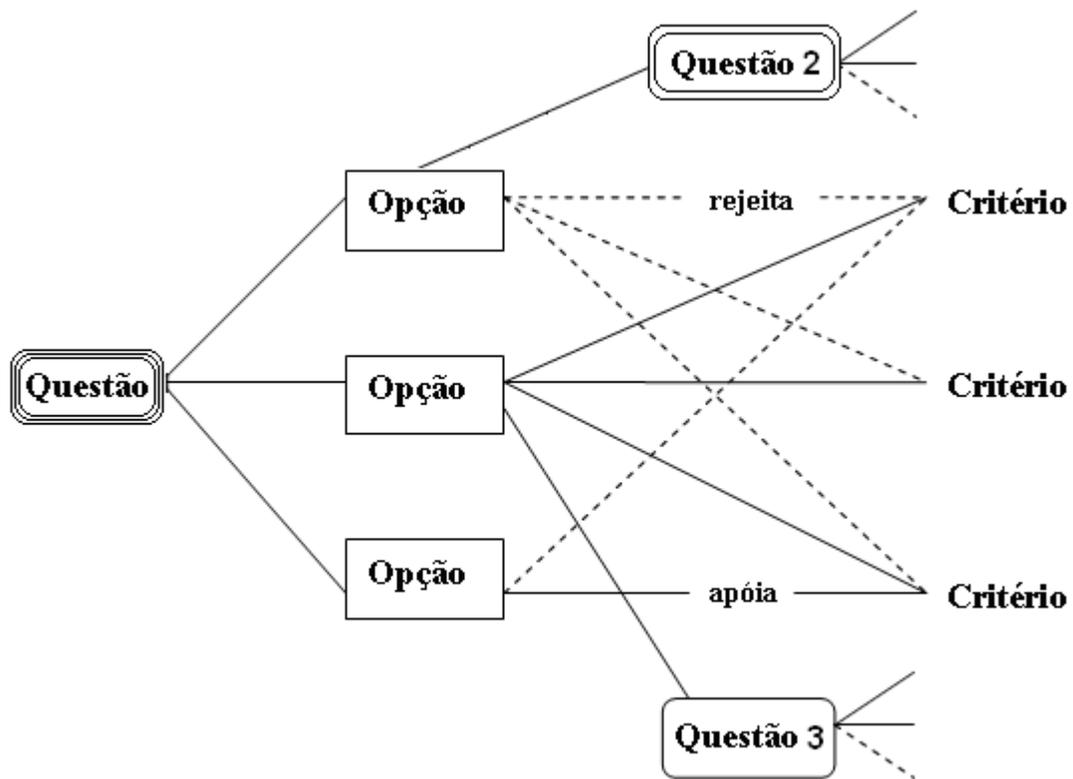
As questões são um dos itens mais importantes da modelagem baseada no QOC, pois é a partir delas que é desencadeado todo o processo da modelagem. Sendo assim, pode-se dizer que a qualidade da modelagem depende da qualidade das questões que dão origem a modelagem. De acordo com Bellotti et. al. (1991), a identificação de boas questões é crítica, porque tais questões podem reestruturar a maneira como os problemas são vistos e definir onde buscar as soluções para estes problemas.

Com base nesta premissa é que se introduziu a representação da importância das questões, pois se acredita que isto facilite a leitura da representação e permita verificar os pontos críticos do projeto e suas decisões relacionadas.

Para simbolizar esta importância de forma diagramática, as questões passam a ser representadas por caixas com bordas arredondadas, sendo que:

- caixas com **borda simples** representam uma questão de importância moderada;
- caixas com **borda dupla** representam uma questão de importância média;
- caixas com **borda tripla** representam uma questão de importância elevada.

A figura 5.1 exibe a representação conceitual do digrama utilizando a representação de importância das questões.



**Figura 5.1:** Exemplo conceitual utilizando a representação de importância das questões.

### 5.1.3 Representação do Peso dos Critérios

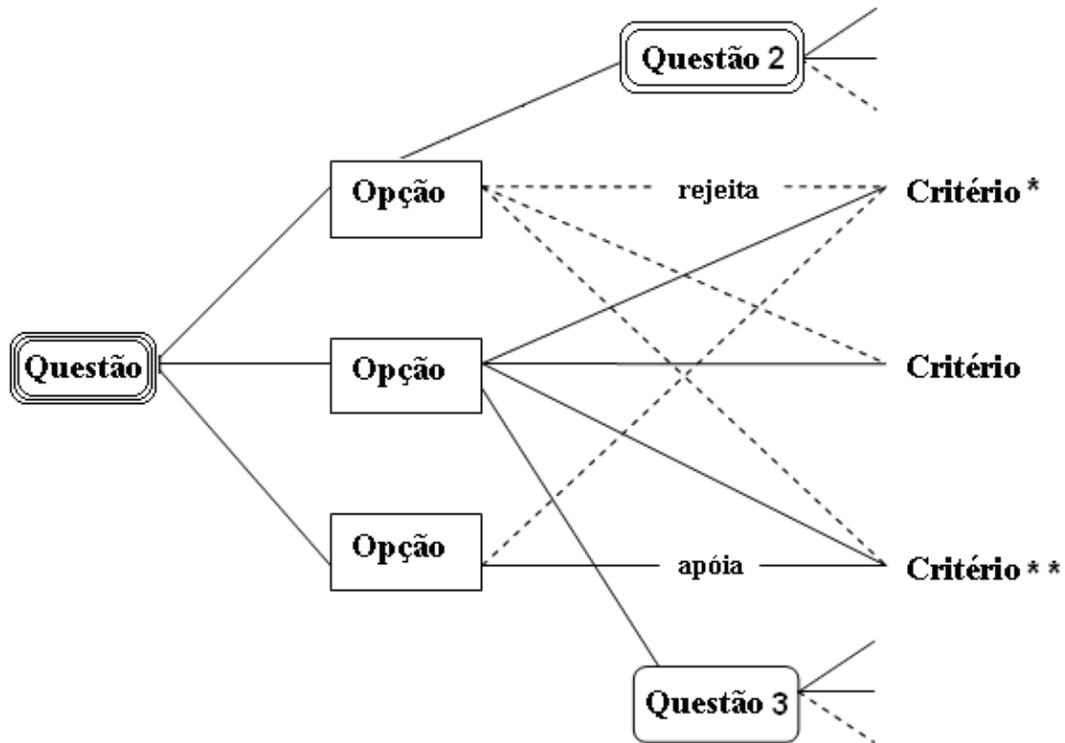
De forma semelhante às questões, os critérios também passaram a ter uma representação de importância, cujo objetivo é facilitar a identificação de quais opções são mais apropriadas para resolver determinada questão.

A utilização de pesos surge como uma ferramenta para ajudar a identificar, dentre os critérios estabelecidos para uma decisão, aqueles que são mais significativos. Sendo assim, introduziu-se a representação do peso de um critério através do uso de asteriscos (\*), de forma que:

- critério **sem asterisco** representa um critério normal, com relevância moderada;
- critério com **um asterisco (\*)** representa um critério importante com relevância média;

- critério com **dois asteriscos (\*\*)** representa um critério muito importante e altamente relevante.

A figura 5.2 exibe a representação conceitual do digrama utilizando a representação de peso para os critérios.



**Figura 5.2:** Exemplo conceitual utilizando a representação de pesos para os critérios.

#### 5.1.4 Descrição Textual

A opção por uma representação diagramática baseia-se no fato de que esta é uma forma mais objetiva e compacta de representação, que ao mesmo tempo possibilita uma visão mais ampla do projeto. Porém, como complemento, pode ser adicionada, a esta, uma descrição textual sucinta que visa explicar a representação dos diagramas. A grande razão para se ter essa descrição é oferecer uma forma de contextualização para quem está analisando a representação e as decisões tomadas, pois somente a representação diagramática pode dificultar a percepção do contexto.

Basicamente, o complemento textual consiste em uma breve descrição das Questões e Critérios classificados com relevância alta ou média. Para isto, sugere-se a utilização de tabelas para fazer a descrição dos itens, como no modelo exibido na Tabela 5.1.

**Tabela 5.1:** Sugestão de tabela para representação textual das principais Questões e Critérios.

<b>Questões de relevância alta</b>	
<b>Questão</b>	<b>Descrição</b>
<b>Questões de relevância média</b>	
<b>Questão</b>	<b>Descrição</b>
<b>Critérios de relevância alta</b>	
<b>Critério</b>	<b>Descrição</b>
<b>Critérios de relevância média</b>	
<b>Critério</b>	<b>Descrição</b>

## 5.2 Aplicação da Proposta

De forma semelhante ao estudo de caso apresentado no capítulo 4, após a definição da proposta, realizou-se um estudo experimental com o objetivo de verificar a aplicabilidade desta proposta em um projeto de software. Nesta fase contou-se com a participação de profissionais de mercado, que participam das decisões tomadas nos projetos em que atuam. Ao longo deste capítulo a palavra experimento será utilizada referindo-se ao estudo realizado para verificar a aplicabilidade da proposta apresentada, porém é importante deixar claro que este estudo não é um experimento no sentido formal da palavra, com hipóteses e controle rígido de variáveis.

### 5.2.1 Perfil dos Participantes

Os experimentos foram realizados por um total de 10 pessoas, sendo que praticamente todas elas possuem experiência em gerência de projetos ou arquitetura de software, além de experiência em outras áreas como análise/projeto, desenvolvimento e testes, como pode ser visto na Tabela 5.2.

**Tabela 5.2:** Tempo de experiência dos participantes do experimento por área.

Participante	Experiência em anos por área de atuação				
	Gerência de projetos	Arquitetura de Software	Desenvolvimento de software	Análise/Projeto de software	Testes de software
1	2		8	4	
2	10		15	15	
3		4	9	7	
4	8				
5	1,5				7
6	3	1			7
7			4	2	
8	5		6	6	7
9	1		10	10	
10	9		20	12	

O perfil dos participantes foi identificado através de um questionário (Apêndice B), que foi preenchido no início do experimento. Informações complementares sobre o perfil das pessoas que contribuíram com esta pesquisa estão disponíveis na Tabela 5.3.

**Tabela 5.3:** Informações complementares sobre o perfil dos participantes do experimento.

Participante	Idade	Já usou outra técnica?	Participa do processo de decisão?
1	26	Sim	Diretamente
2	41	Não	Diretamente
3	30	Não	Diretamente
4	34	Não	Indiretamente
5	26	Sim	Diretamente
6	27	Sim	Diretamente
7	26	Não	Indiretamente
8	28	Sim	Diretamente
9	34	Sim	Diretamente
10	32	Não	Indiretamente

Pode-se observar que a faixa etária dos participantes variou entre 26 e 41 anos, o que caracteriza uma população com uma grande diversidade de conhecimento e experiência em projetos de software, seja em termos de tecnologias utilizadas ou metodologia de gerenciamento de projetos. Acredita-se que este fato contribuiu para uma maior riqueza no resultado do experimento, pois são pessoas com experiências bem diferenciadas.

Outro fato a ser considerado é que metade dos participantes disse nunca ter usado nenhuma técnica para documentar decisões. Da outra metade, que já utilizou, a maioria mencionou atas de reuniões ou *e-mails* para documentação, sendo que ainda foram citadas outras técnicas como mapas mentais (*mind maps*) e diagramas espinha de peixe, além de outros modelos específicos a determinados projetos.

Ao mesmo tempo em que as informações acima demonstram que não é muito comum se documentar decisões, considerando que a metade da população observada nunca utilizou nenhuma técnica para isso, também é possível perceber que existe a preocupação de se manter estas informações e, inclusive, há iniciativas neste sentido, seja com documentação puramente textual como atas e *e-mails* ou então com a utilização de algum tipo de representação.

### 5.2.2 Dinâmica do Experimento

Em termos gerais, o experimento consistiu na modelagem, pelos participantes, de um sistema de software fictício, utilizando a abordagem QOC \*. O objeto da modelagem foi um sistema para gerenciamento de projetos de pesquisa acadêmicos, composto por três módulos: 1) Gerência de RH; 2) Gerência de Recursos financeiros e imobilizados; 3) Gerência dos projetos propriamente ditos, como publicações e resultados. Para não tornar o experimento algo muito complexo e demorado, foi solicitada a modelagem de somente uma parte do projeto, sendo que a parte escolhida foi o gerenciador de currículos, que pertence ao módulo de Gerência de RH. Maiores detalhes sobre o projeto podem ser encontrados no Apêndice C.

A intenção de realizar o experimento com profissionais que trabalham em projetos de software e participam das decisões nestes projetos foi obter o *feedback* destas pessoas a respeito da abordagem QOC \*, sendo que estes profissionais são o público alvo da proposta e podem, desta forma, avaliá-la em termos de aplicabilidade e relevância, considerando a realidade de um projeto de software.

Para a realização do experimento, foi preparado um roteiro contendo a contextualização do assunto, a descrição da abordagem QOC \*, um exemplo de modelagem com QOC \*, a descrição do sistema para o qual deveria ser feita a análise das possíveis decisões utilizando a nova abordagem e um questionário para avaliar a experiência proporcionada ao usuário e as dificuldades relacionadas ao desenvolvimento da modelagem. Além disso, os participantes assinaram um termo de consentimento no qual cada participante manifesta estar de acordo com a divulgação dos resultados da pesquisa, sendo que o anonimato dos mesmos é garantido. Os documentos com o roteiro do experimento e o termo de consentimento estão disponíveis nos Apêndices C e D, respectivamente.

Devido à dificuldade para agendar horário com todos os participantes, optou-se por duas modalidades de execução do experimento: 1) com monitoramento, em que os participantes foram assistidos e tiveram liberdade para questionar e tirar dúvidas durante a realização do experimento; 2) individual, em que o participante realizou o experimento sozinho, sem assistência, apenas baseado no documento com o roteiro.

Um ponto interessante da segunda opção foi justamente verificar a dificuldade no entendimento da proposta e no desenvolvimento da modelagem, visto que apenas uma descrição textual foi oferecida, sem maiores explicações.

De maneira geral, a realização do experimento propiciou a obtenção de informações importantes sobre a abordagem proposta, como aplicabilidade da mesma, dificuldade de entendimento, além de vantagens da sua utilização. Na seção seguinte é apresentada uma discussão sobre a proposta e sobre os resultados do experimento.

### 5.3 Discussão

A partir da definição do perfil do público alvo, partiu-se a procura de pessoas que atendessem ao perfil necessário e pudessem participar do experimento. Uma das principais dificuldades encontradas para a aplicação deste experimento foi encontrar pessoas com o perfil desejado e com disponibilidade para participar. No total, foram convidadas 35 pessoas, sendo que destas, 10 tiveram disponibilidade para participar da pesquisa. A Tabela 5.4 apresenta um sumário das informações obtidas com o experimento, sendo que estas são comentadas na seqüência. A título de conhecimento, somente uma pessoa que participou do estudo de caso apresentado no capítulo 4 também participou novamente desta fase, sendo que para os outros participantes o contexto da proposta foi algo novo.

**Tabela 5.4:** Resultados do experimento.

<b>Participante</b>	<b>Modalidade de execução</b>	<b>Duração (min)</b>	<b>Nível de dificuldade da proposta</b>	<b>Util em quais fases de um projeto</b>
<b>1</b>	Individual	35	Baixo	Análise e Projeto
<b>2</b>	Individual	80	Baixo	Análise e Projeto
<b>3</b>	Monitorado	40	Baixo	Arquitetura, Análise e Projeto
<b>4</b>	Monitorado	30	Médio	Análise
<b>5</b>	Monitorado	120	Médio	Análise
<b>6</b>	Individual	50	Alto	Análise
<b>7</b>	Monitorado	40	Baixo	Análise
<b>8</b>	Individual	45	Baixo	Análise e Projeto
<b>9</b>	Individual	60	Baixo	Análise
<b>10</b>	Individual	50	Médio	Análise

Um ponto observado na realização do experimento foi o tempo de duração para cada participante, ou seja, quanto tempo cada pessoa levou para entender a técnica e realizar a modelagem das decisões envolvidas no sistema fictício apresentado. O tempo médio necessário foi de 55 minutos, sendo o maior tempo de duração, 2 horas e o menor, 35 minutos. É importante destacar que o sistema apresentado para ser modelado é relativamente simples, pois o objetivo não é o desenvolvimento de uma modelagem complexa, mas sim identificar a dificuldade de entendimento e aplicação da abordagem QOC \*.

Com relação ao entendimento da abordagem e da notação do QOC \*, 90% dos entrevistados consideraram de médio a fácil, achando os elementos bem definidos, intuitivos e expressivos. Em dois casos houve relato de dificuldade no entendimento em um primeiro momento, sendo que esta dificuldade acabou sendo superada e o entendimento aconteceu durante a construção do diagrama.

Sobre a aplicabilidade da proposta em projetos de software, os 10 entrevistados consideram a mesma aplicável, principalmente pelo fato da abordagem instigar o levantamento de questões e a busca de opções para atender estas questões, o que incentiva a discussão. Ao mesmo tempo, a abordagem incentiva a formalização do que está sendo discutido, proporcionando, desta forma, um histórico das questões e das opções analisadas, bem como dos critérios que levam a aceitação ou não de determinada opção.

No entanto, percebeu-se certa preocupação com relação à utilização da abordagem proposta em projetos grandes e complexos, no sentido de que a representação possa se tornar confusa e muito complexa, o que dificultaria o entendimento da modelagem e consequentemente sua utilização. Além disso, foi levantada a necessidade de uma ferramenta para ajudar na aplicação do QOC \*, de forma a facilitar a criação dos diagramas e possibilitar uma forma padronizada de representação, bem como possibilitar a organização e a recuperação das modelagens.

Além da aplicabilidade da abordagem, os participantes do experimento foram questionados a respeito da fase do projeto em que este tipo de modelagem melhor se adaptaria, sendo que, por unanimidade, foi citada a fase inicial de um projeto, ou seja, na investigação e análise de requisitos. Outras fases também foram citadas, como o planejamento da arquitetura e o projeto do software.

Uma consideração levantada durante o experimento, e que também é considerada e destacada neste trabalho, é que esta abordagem deve ser utilizada nas principais e mais importantes decisões de um projeto, e não em toda e qualquer decisão. O uso de forma exhaustiva da proposta seria bastante difícil considerando a realidade de um projeto de desenvolvimento de software.

A partir das entrevistas, observou-se como principais vantagens da proposta apresentada a possibilidade de se manter e reaproveitar os problemas identificados e as opções avaliadas em projetos de software, pois estas informações são úteis e importantes para a continuidade de um projeto. Outro aspecto a se considerar é que essas informações podem, também, vir a ser utilizadas em outros projetos, pois muitas decisões acontecem diante de cenários recorrentes, logo, o histórico das decisões e suas razões, desde que acessível, pode ser útil para gerenciar o conhecimento existente em um projeto.

Além da necessidade de uma ferramenta para realizar a modelagem utilizando o QOC \*, observou-se a necessidade de uma ferramenta que permita a pesquisa em uma base de conhecimento que contenha as informações sobre as decisões de diversos projetos e possibilite o relacionamento entre estas informações, automatizando assim o processo de reaproveitamento das informações armazenadas.

De maneira geral, acredita-se que o resultado da utilização do QOC \* foi positivo, considerando-se que as pessoas conseguiram entender a abordagem apresentada com relativa facilidade e vislumbrar pontos positivos na sua utilização nos projetos em que trabalham. Como uma proposta nova, com certeza há muitos pontos que podem ser aprimorados, tornando-a, assim, mais atraente e completa.

É importante também considerar que existem várias técnicas, com propósitos diferentes, que podem ser utilizadas em projetos de software. Devido a natureza destes projetos, muitas vezes a inserção de novas técnicas é um desafio, pois provoca um impacto no projeto, o que nem sempre é aceitável.

## 5.4 Refinamento da Proposta

Após a realização do experimento e da análise dos resultados, notou-se que algumas características ainda poderiam ser alteradas ou adicionadas, com a intenção de tornar a proposta mais interessante e completa. Estas características são detalhadas nas subsecções seguintes, e necessitam de uma nova validação, o que está sendo proposto como trabalho futuro.

### 5.4.1 Ampliação da Descrição Textual

A tabela para descrição textual apresentada na proposta original abrange somente as questões e critérios considerados de relevância média ou alta. Porém, julgou-se interessante também descrever as opções analisadas além de todas as questões e critérios, incluindo as de relevância mais baixa. Através da descrição textual de todos os itens que compõe a modelagem, acredita-se que a contextualização do projeto fique mais completa e facilite a leitura da representação diagramática. Uma sugestão de tabela para descrever os itens da representação diagramática é apresentada na Tabela 5.5.

**Tabela 5.5:** Sugestão de tabela para representação textual das Questões, Opções e Critérios.

<b>Questões</b>	
<b>Questão</b>	<b>Descrição</b>
Questão Q1	Breve descrição da questão Q1
Questão Q2	Breve descrição da questão Q2
<b>Opções</b>	
<b>Opção</b>	<b>Descrição</b>
Opção O1	Breve descrição da opção O1
Opção O2	Breve descrição da opção O2
<b>Critérios</b>	
<b>Critério</b>	<b>Descrição</b>
Critério C1	Breve descrição do critério C1
Critério C2	Breve descrição do critério C2

### 5.4.2 Tipos de Decisão

É importante lembrar que é inviável avaliar e representar todo tipo de questão envolvida em um projeto de software, sendo que isso geraria uma complexidade muito grande e tornaria impossível armazenar e consultar as informações relacionadas às decisões tomadas.

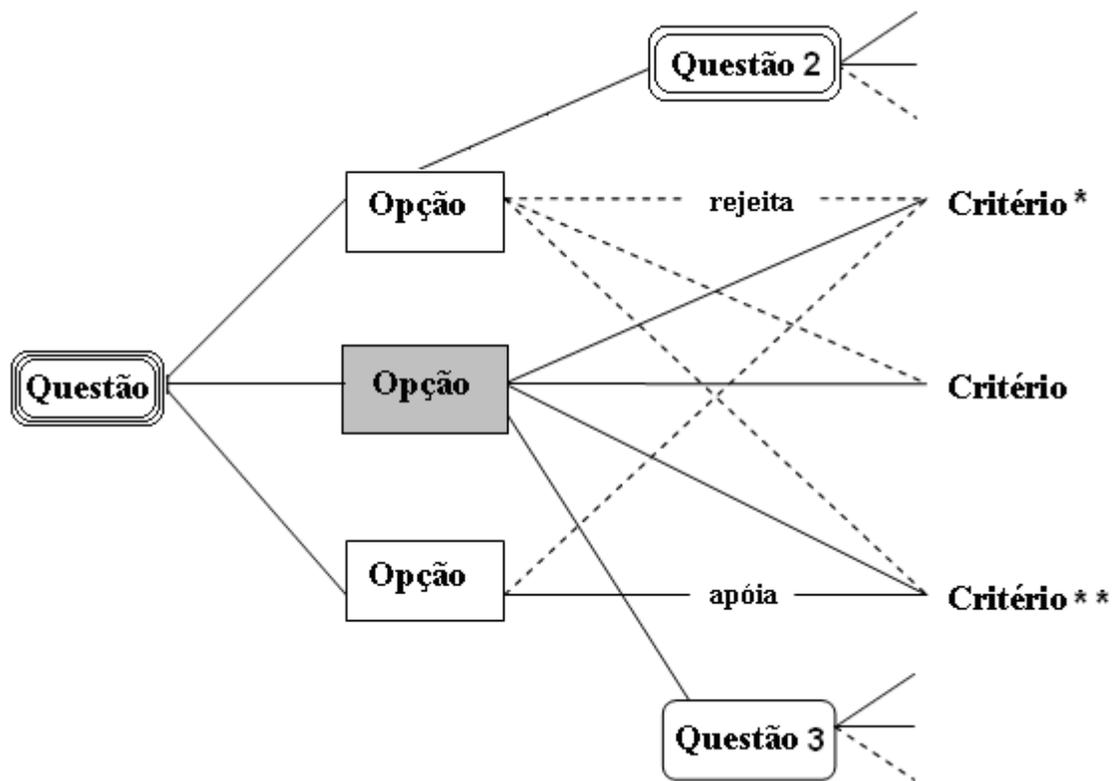
Sendo assim, para efeitos de representação é necessário considerar somente as questões mais importantes de todo o projeto, cujo impacto e risco são relevantes e cujas decisões caracterizam determinado projeto. Abaixo são listados alguns tipos de decisões consideradas relevantes através do estudo e das entrevistas realizadas:

- decisões sobre regras de negócio;
- decisões de projeto/arquitetura do software;
- decisões sobre a arquitetura de desenvolvimento do projeto;
- decisões que impactam e ameaçam o cronograma do projeto.

### 5.4.3 Identificação da Decisão Tomada

Apesar da abordagem em questão ter como objetivo manter tanto as decisões tomadas quanto às opções que deixaram de ser escolhidas por determinada razão, é importante ter alguma forma de identificar qual das alternativas avaliadas foram realmente escolhidas. Para isso, optou-se por representar a opção escolhida através de uma caixa preenchida, diferentemente das outras opções que utilizam caixa sem preenchimento, destacando assim as opções selecionadas e aplicadas no projeto.

A figura 5.3 exibe a representação diagramática da opção selecionada, lembrando que apesar do exemplo da figura destacar somente uma opção, é perfeitamente possível que mais de uma opção seja destacada em um mesmo diagrama, dependendo, é claro, das decisões que foram tomadas. Acredita-se que a representação das decisões tomadas facilite a leitura do diagrama, tornando direta a percepção de quais foram as opções consideradas e quais não foram utilizadas.



**Figura 5.3:** Exemplo conceitual com a representação que destaca a opção escolhida.

## 6 CONSIDERAÇÕES FINAIS

Neste capítulo são apresentadas as considerações finais sobre o desenvolvimento da pesquisa, em termos das principais conclusões e contribuições alcançadas, além de uma breve descrição de possíveis trabalhos futuros, que possam dar continuidade a proposta apresentada nesta dissertação.

### 6.1 Conclusões

Considerando-se a grande quantidade de conhecimento necessário e existente em projetos de software, o presente trabalho apresentou uma proposta para representar o conhecimento envolvido no processo de tomada de decisão deste tipo de projeto. Esta proposta abrange a representação não somente das alternativas escolhidas, mas sim de todas as alternativas avaliadas para resolver os principais problemas de decisão presentes em projetos de desenvolvimento de software. A representação proposta pelo QOC \*, baseada na abordagem QOC, estimula a definição das principais questões existentes em um projeto e a análise das alternativas que possam vir a ser escolhidas para suprir as necessidades levantadas em cada questão.

Ao longo deste trabalho, abordou-se conceitos que estão diretamente relacionados à proposta apresentada, como Gestão do Conhecimento e *Design Rationale*. A Gestão do Conhecimento consiste em uma disciplina mais ampla e genérica, que pode ser aplicada as mais variadas áreas possíveis, mas que busca, de maneira geral, capturar e democratizar o acesso ao conhecimento. Dentro do contexto desta dissertação, identificou-se o *Design Rationale* como uma ferramenta mais específica, mas que pode ser utilizada para gerenciar o conhecimento, neste caso o conhecimento relacionado às decisões de projeto.

Uma vez definidos os conceitos envolvidos nesta pesquisa, foram realizadas entrevistas com profissionais que atuam em projetos de desenvolvimento de software, com o objetivo de expor o tema da pesquisa e identificar algumas necessidades destes profissionais em termos de documentação das decisões tomadas ao longo do projeto. Durante as entrevistas, foram

apresentadas algumas possibilidades de documentação (com exemplos), e discutiu-se a aplicabilidade destas no dia-a-dia de um projeto. Ainda durante as entrevistas, foram observadas várias considerações feitas pelos entrevistados, com relação às carências e às sugestões para documentar as decisões e as razões das decisões de projeto.

A partir da análise do resultado das entrevistas, elaborou-se uma proposta baseada em *Design Rationale*, a qual agrega algumas das sugestões feitas pelos entrevistados e aspectos observados nos estudos realizados. Assim que a proposta foi definida, foi necessária novamente a participação de profissionais para ajudar a validar a mesma. Desta vez os participantes tiveram que modelar as decisões envolvidas em um sistema fictício utilizando a abordagem QOC \*, e então responder a um questionário, cujo objetivo é o *feedback* dos participantes a respeito da proposta.

De uma maneira geral, a partir dos resultados da pesquisa, acredita-se que a proposta permite a criação dos quatro padrões de conhecimento descritos na subseção 2.1.2 deste trabalho. O primeiro é a externalização, pois a representação proposta possibilita tornar o conhecimento tácito das pessoas que participam do processo de decisão em um conhecimento explícito e acessível a outras pessoas. O segundo padrão é a socialização, pois através das discussões para a construção da representação, utilizando o modelo proposto, as pessoas podem absorver o conhecimento entre si. O terceiro padrão é a internalização, que consiste na criação de um conhecimento tácito a partir do conhecimento explícito possibilitado pela representação formal do QOC \*. O quarto padrão é a combinação, criado através do reaproveitamento do conhecimento documentado em um projeto para a geração de uma nova documentação em outro projeto, obtendo-se assim conhecimento explícito a partir de conhecimento explícito.

Um outro aspecto observado, é que a atividade de desenvolvimento de software é uma atividade diretamente dependente do conhecimento, de forma que muitas vezes este conhecimento está concentrado em apenas algumas pessoas, logo, a saída de um colaborador chave pode significar um impacto muito grande no projeto. Através da utilização de abordagens como o QOC \* é possível se manter aspectos chave do projeto e principalmente, o porquê destes aspectos, o que com certeza facilita a disseminação do conhecimento, a curva de aprendizagem de um novo colaborador e o entendimento global do projeto.

Mais do que uma forma ou processo de representação, acredita-se que a proposta apresentada sirva como um estímulo ao questionamento e a discussão sobre as melhores alternativas para a resolução de um problema. De acordo com Hussain et. al. (2004), a cultura é

um dos fatores chave para a Gestão do Conhecimento, ou seja, a cultura de se compartilhar o conhecimento adquirido e assim gerar mais conhecimento. Isto é exatamente o que se espera com este trabalho, ou seja, estimular a cultura de questionar e discutir a respeito das possíveis soluções e armazenar de forma explícita o conhecimento adquirido para que o mesmo seja compartilhado, aprimorando assim processos e obtendo melhores resultados.

## 6.2 Trabalhos Futuros

Além dos aspectos cobertos por este trabalho, identificaram-se alguns outros aspectos que podem fazer parte de trabalhos futuros, e que são importantes para a evolução e validação da proposta apresentada. Estes aspectos são:

- **Reaplicação da proposta incluindo as características detalhadas na seção 5.4:** considerando que estas características não fizeram parte do experimento, é necessário validá-las, obtendo-se, assim, o *feedback* a partir da última versão da proposta.
- **Análise do impacto da aplicação da proposta em um projeto de software real:** utilização da abordagem proposta em um ou mais projetos de software reais, para que seja feita uma análise do impacto causado, bem como dos reais benefícios gerados.
- **Desenvolver uma ferramenta de software para facilitar a modelagem:** desenvolvimento de uma ferramenta de software que facilite a criação dos diagramas (modelagem gráfica e textual), bem como sua organização e acesso.
- **Desenvolver um modelo computacional:** desenvolvimento de um modelo computacional e de uma ferramenta que transforme a modelagem gráfica e textual neste modelo computacional, abrindo assim a possibilidade de integração com outras ferramentas e a realização de inferências a partir da base de dados de decisões e alternativas avaliadas com o QOC \*. Este tipo de integração com certeza facilitaria o reaproveitamento do conhecimento capturado com a utilização da abordagem proposta.

## REFERÊNCIAS

- AHERN, D.; CLOUSE, A.; TURNER, R. **CMMI Distilled: A Practical Introduction to Integrated Process Improvement**. 2nd edition: Addison Wesley, 2003. 336 p.
- BASKERVILLE, R.; PRIES-HEJE, J. Knowledge Capability and Maturity in Software Management, **ACM SIGMIS Database**, vol. 30, nº 2, March 1999. pp 26-43.
- BELLOTTI, V.; MACLEAN, A.; and MORAN, T. What makes a Good Design Question? **SIGCHI Bulletin**, vol. 23, nº 4, October 1991. pp 80-81.
- BHIRUD, S.; RODRIGUES, L.; DESAI, P. Knowledge Sharing Practices In Km: A Case Study In Indian Software Subsidiary, **Journal of Knowledge Management Practice**, December 2005. Disponível em: <<http://www.tlinc.com/jkmp.htm>>. Acesso em: 10/05/2006.
- BIRK, A.; SURMANN, D.; ALTHOFF, K. **Applications of Knowledge Acquisition in Experimental Software Engineering**, In: Proceedings of the 11th European Workshop on Knowledge Acquisition, Modeling and Management, May 1999. pp. 67-84.
- CHRISSIS, M. B.; KONRAD, M.; SHRUM, S. **CMMI Guidelines for Process Integration and Product Improvement**, Addison-Wesley, 2003. 688 p.
- DAVENPORT, T.; PRUSAK, L. **Conhecimento Empresarial: como as organizações gerenciam seu capital intelectual**, Rio de Janeiro: Campus, 4ª edição, 1998. 237 p.
- DESOUZA, K. C. Barriers to Effective Use of Knowledge Management Systems in Software Engineering, **Communications of the ACM**, vol. 46, nº 1, January 2003. pp 99-101.
- DINGSOYR, T.; ROYRVIK, E. **An Empirical Study of an Informal Knowledge Repository in a Medium-Sized Software Consulting Company**, In: Proceedings of the 25th International Conference on Software Engineering, May 2003. pp 84-92.
- DRUCKER, P. **Sociedade Pós-Capitalista**, São Paulo: Pioneira, 6ª edição, 1997. 186 p.
- GARVIN, D. A. **Construindo a Organização que Aprende**, In: Harvard Business review. Gestão do Conhecimento. 4ª ed. Rio de Janeiro: Campus; 2000. pp 50-81.

- GUPTA, B.; IYER, L. S.; ARONSON, J. E. Knowledge Management: Practices and Challenges, **Industrial Management and Data Systems**, vol.100, nº 1, 2000. pp 17-21.
- HOLSAPPLE, C. W.; WHINSTON, A. B. **Decision Support Systems: A Knowledge Based Approach**, West Publishing, 1996.
- HORNER, J.; ATWOOD, M. E. **Design rationale: the rationale and the barriers**. In: Proceedings of the 4th Nordic Conference on Human-computer interaction, Oslo, Norway, October, 2006. pp 341-350.
- HUSSAIN, F.; LUCAS, C.; ALI, M. A. Managing Knowledge Effectively, **Journal of Knowledge Management Practice**, May 2004. Disponível em: <<http://www.tlinc.com/articl66.htm>>. Acesso em: 10/05/2006.
- KUNZ, W.; RITTEL, H. **Issues as elements of information systems**. Center for Urban and Regional Development, University of California Berkley, 1970. Disponível em: <<http://www-iurd.ced.berkeley.edu/pub/WP-131.pdf>>. Acesso em: 12/11/2007.
- LARA, S.M.A. **Um Suporte à captura informal de design rationale**, Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação de São Carlos, Universidade de São Paulo, 130f, 2005.
- LAWTON, G. Knowledge Management: Ready for Prime Time? **Computer**, vol. 34, nº 2, 2001, pp. 12-14.
- LEE J. **Extending the Potts and Bruns model for recording design rationale**. In: Proceedings of 13th International Conf. on Software Engineering, Austin, Texas, May 1991. pp 114-125.
- LEE, J. Design Rationale Systems: Understanding the Issues, **IEEE Expert**, Vol. 12, nº 3, May 1997, pp. 78-85.
- LEE J; LAI, K. What's in Design Rationale?, In **Design Rationale: Concepts, techniques, and Use**, Moran and Carroll eds., Lawrence Erlbaum, 1996. pp 251-280.
- MACLEAN, A.; YOUNG, R. M.; BELLOTTI, V.; MORAN, T. Questions, Options, and Criteria: Elements of Design Space Analysis. In **Design Rationale: Concepts, Techniques, and Use**, T. P. Moran and J. M. Carroll, Eds. Lea Computers, Cognition, And Work Series. Lawrence Erlbaum Associates, Mahwah, NJ. pp 53-105.

- MATHI, K. **Key Success Factors for Knowledge Management**, 102 p, Master Thesis, University of Applied Sciences, Fh Kempten, Germany, 2004.
- MCCALL, R. J. PHI: A Conceptual Foundation for Design Hypermedia. **Design Studies**, vol. 12, nº 1, 1991, pp 30-41.
- MCKERLIE, D.; MACLEAN, A. **QOC in Action: Using Design Rationale to Support Design**, In Proceedings of the INTERACT '93 e CHI '93 Conference on Human Factors in Computing Systems, Amsterdam, The Netherlands, April 1993.
- MEDEIROS, A. P. **Kuaba: Uma Abordagem para Representação de Design Rationale para o Reuso de Designs baseado em Modelos**, Pontifícia Universidade Católica do Rio de Janeiro, 149 f. Tese de Doutorado, Rio de Janeiro, março, 2006.
- MORAN, T. P.; CARROLL, J. M.; LEE J.; LAI, K. **Overview of Design Rationale**, in 'Design Rationale: Concepts, techniques, and Use', Moran and Carroll eds., Lawrence Erlbaum, 1996.
- NATALI, A.; FALBO, R. **Knowledge Management in Software Engineering Environments**, In: Proceedings of SBES, 2002. pp 238-253.
- NONAKA, I. "The Knowledge Creating Company", **Harvard Business Review**, vol. 69, 1991. pp 96-104.
- PREECE, J.; ROGERS, Y.; SHARP, H.; BENYON, D.; HOLLAND, S.; CAREY, T. **Human-Computer Interaction**. Harlow: Addison-Wesley, 1994.775 p.
- RUS, I.; LINDVALL, M; SINHA, S. S. **Knowledge Management in Software Engineering - A State-of-the-Art-Report**, The University of Maryland, 2001.
- SHIPMAN, F; MCCALL, R. Integrating different perspectives on design rationale: Supporting the emergence of design rationale from design communication. **Artificial Intelligence for Engineering Design, Analysis and Manufacturing**, vol. 11, nº 2, April 1997, pp 141-154.
- SHUM, S. B.; HAMMOND, N. Argumentation-Based Design Rationale: What Use at What Cost?, **International Journal of Human-Computer Studies**, vol 40, nº 4, April 1994, pp 603-652.
- SIVAN, Y.Y. **Nine Keys To A Knowledge Infrastructure**. Harvard University, March 2001.

- SOARES, F. F. **Fatores de Sucesso Para a Adoção e Aplicação de Gestão do Conhecimento: O Caso de Uma Organização de Desenvolvimento de Software**, Dissertação de Mestrado, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2005. 174 p.
- SOUZA, C. R. B.; SANTOS, D. B.; WAINER, J.; DIAS, K. **A model tool for semi-automatic recording of design rationale in software diagrams**. In: Proceedings of the String Processing and Information Retrieval Symposium, Washington, September 1999. pp 306-313.
- STUMPF, S. **Argumentation-based Design Rationale - the sharpest tools in the box**. Disponível em: <<http://www.cs.ucl.ac.uk/staff/S.Stumpf/Reports/IN9801.html>>. Acesso em 15/10/2006.
- WALZ, D.; ÉLAN, J. J.; CURTIS, B. Inside a Software Design Team: Knowledge Acquisition, Sharing and Integration, **Communications of the ACM**, vol. 36, October 1993. pp 63-77.
- ZACK, M.H. Developing a Knowledge Strategy, **California Management Review**, vol. 41, 1999. pp 63-77.

## APÊNDICE A

### Questões da entrevista

Apesar das entrevistas terem sido realizadas através de discussão livre, as questões abaixo representam o roteiro seguido para se obter as informações utilizadas neste trabalho.

1. Você acredita que as informações de um projeto possam ser úteis para outro?
  - Já usou isto na prática?
2. O tempo usado para documentar compensa o impacto no cronograma?
3. Você já utilizou alguma técnica para representar decisões de projeto?
  - Quais?
  - Como foi a experiência?
  - Utilizaria novamente?
4. Qual sua visão com relação a documentação de projetos e o histórico de decisões?
  - Acha útil e viável?
  - Está disposto a usar isso no dia-a-dia?
5. Que informações você acha importantes e úteis neste tipo de documentação?
6. Qual a melhor saída para representação deste tipo de informação? Textual ou diagramática?

### Apresentação das modelagens

7. O que você acha das modelagens apresentadas?
  - O que está faltando?
  - O que é útil?
  - O que acha necessário melhorar?
8. O que você acha da carga de trabalho envolvida para adotar algum destes modelos?
9. Como você vê a utilização de alguma destas técnicas no dia-a-dia?

## APÊNDICE B

### QOC \* Questionário Perfil

Data: \_\_ / \_\_ / \_\_\_\_

Nome: \_\_\_\_\_ Idade: \_\_\_\_ anos

Sexo: ( ) Masculino ( ) Feminino

Profissão: \_\_\_\_\_

1. Tem experiência em:

- ( ) Desenvolvimento. Há quanto tempo: \_\_\_\_\_
- ( ) Análise/Projeto de software. Há quanto tempo: \_\_\_\_\_
- ( ) Arquitetura de software. Há quanto tempo: \_\_\_\_\_
- ( ) Gerência de projeto. Há quanto tempo: \_\_\_\_\_
- ( ) Teste de Software. Há quanto tempo: \_\_\_\_\_

2. Qual função você exerce atualmente:

\_\_\_\_\_

3. Você utiliza ou já utilizou alguma técnica para documentação de decisões? Se sim, qual?

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

4. Você participa do processo de tomada de decisão no seu trabalho:

- ( ) Diretamente
- ( ) Indiretamente
- ( ) Não participo.

## APÊNDICE C

### QOC \*

#### C.1 Introdução

Este experimento busca validar e obter *feedback* sobre a aplicação de uma técnica estendida de *Design Rationale*, cujo objetivo é identificar e armazenar o conhecimento envolvido nas decisões tomadas durante o desenvolvimento de um projeto de software. A seguir serão apresentadas as informações para o desenvolvimento da modelagem utilizando esta técnica.

#### C.2 Descrição da Técnica

A técnica de modelagem consiste basicamente em questões, opções e critérios. As questões representam os problemas de decisão enfrentados em um projeto; as opções buscam oferecer alternativas para resolver as questões levantadas e, os critérios representam os prós e contras de cada opção.

As questões são representadas por caixas com bordas arredondadas, sendo que:

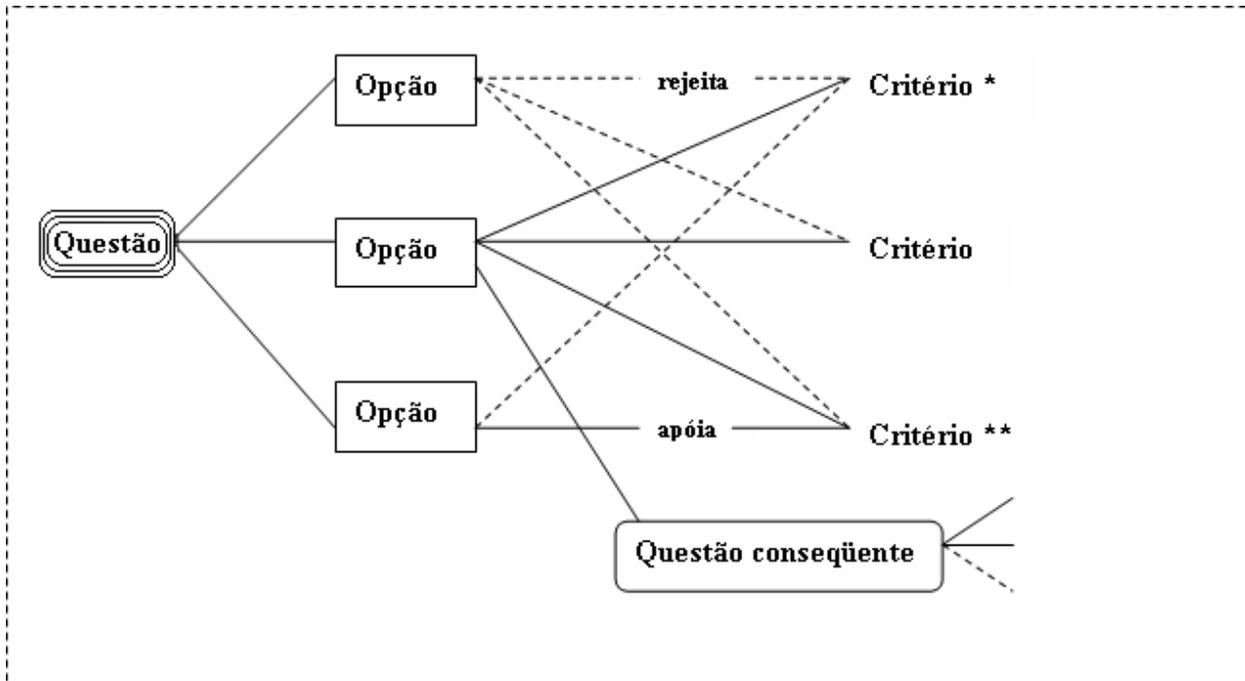
- Caixas com borda simples representam uma questão de relevância moderada;
- Caixas com borda dupla representam uma questão de relevância média.
- Caixas com borda tripla representam uma questão de importância elevada.

As opções são representadas por caixas simples e não possuem representação de peso ou importância. Além disso, as opções são ligadas aos critérios por linhas contínuas em caso de critérios favoráveis e por linhas pontilhadas em caso de critérios desfavoráveis.

Os critérios por sua vez possuem a seguinte representação:

- Sem asterisco: representa um critério normal, com relevância moderada.
- Com um asterisco (\*): representa um critério importante com relevância média.
- Com dois asteriscos (\*\*): um critério muito importante e altamente relevante.

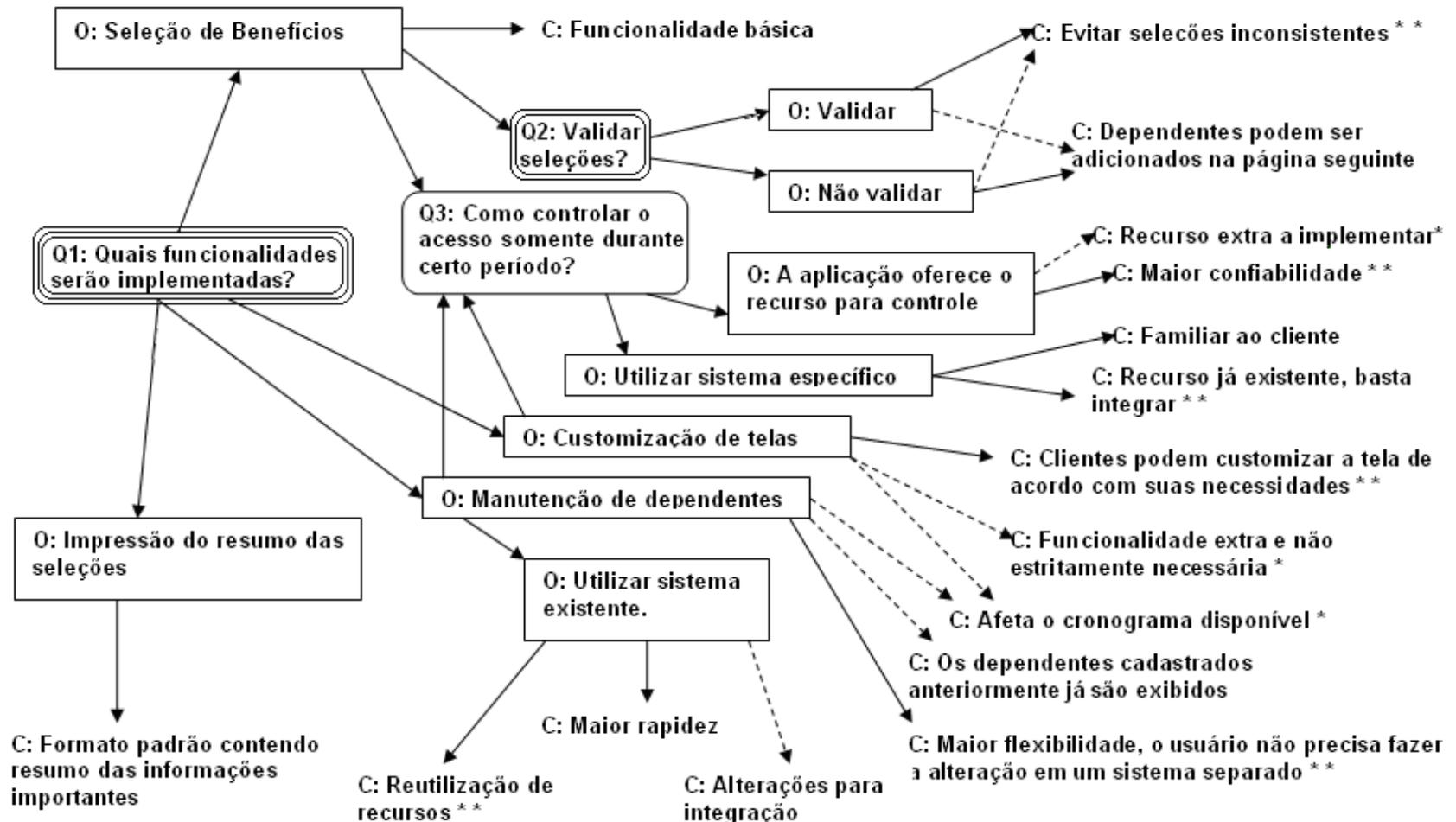
Além da representação diagramática, pode-se utilizar uma sucinta descrição textual das principais questões e critérios definidos no diagrama, através de uma tabela, como a apresentada na descrição conceitual da abordagem. A descrição textual pode ser útil para contextualizar quem está lendo a modelagem. A figura e a tabela abaixo apresentam o modelo conceitual da abordagem QOC \*.



As tabelas a seguir são usadas para descrição da importância das Questões e Critérios principais.

<b>Questões de relevância alta</b>	
<b>Questão</b>	<b>Descrição</b>
<b>Questões de relevância média</b>	
<b>Questão</b>	<b>Descrição</b>
<b>Critérios de relevância alta</b>	
<b>Critério</b>	<b>Descrição</b>
<b>Critérios de relevância média</b>	
<b>Critério</b>	<b>Descrição</b>

### C.3 Exemplo de Modelagem



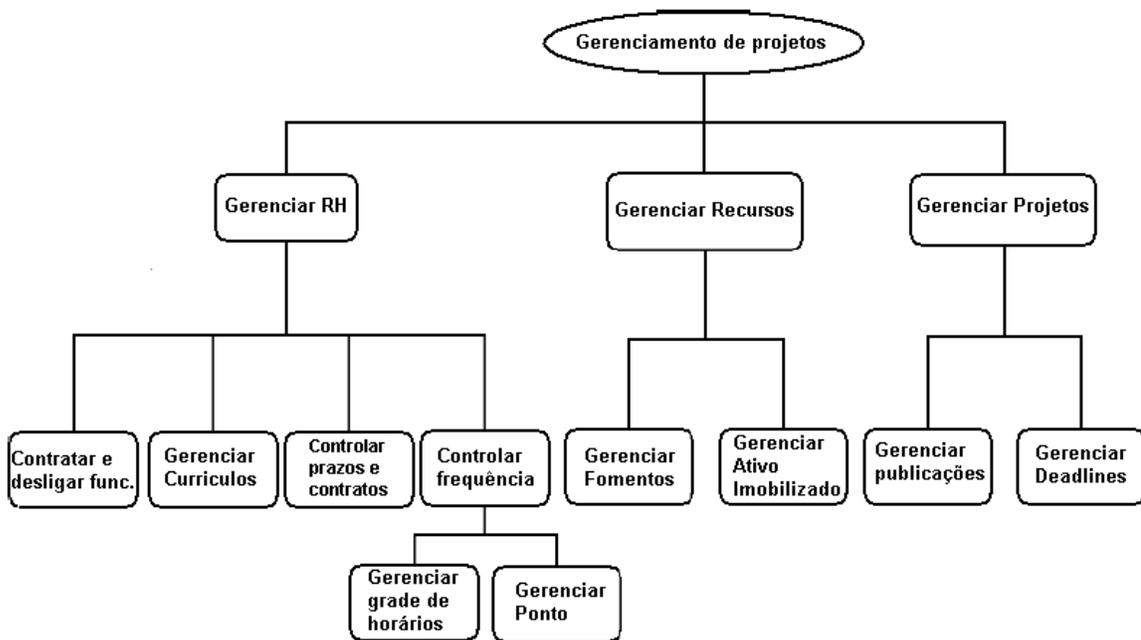
As questões e os critérios especificados abaixo são os que foram considerados mais relevantes de acordo com o diagrama acima.

<b>Questões de relevância alta</b>	
<b>Questão</b>	<b>Descrição</b>
<b>Q1: Quais funcionalidades serão implementadas?</b>	Funcionalidades necessárias e que definirão o escopo do projeto, bem como sua complexidade e esforço de desenvolvimento.
<b>Questões de relevância média</b>	
<b>Questão</b>	<b>Descrição</b>
<b>Q2: Validar seleções?</b>	Afeta o fluxo de navegação entre as páginas e permite manter a consistência na seleção de acordo com as características do usuário.

<b>Crítérios de relevância alta</b>	
<b>Critério</b>	<b>Descrição</b>
<b>C: Evitar seleções inconsistentes * *</b>	Visa manter a consistência das seleções evitando a escolha de opções não compatíveis com o usuário.
<b>C: Recurso já existente, basta integrar * *</b>	A reutilização de recursos existentes, desde que adequada à realidade do projeto, deve ser fortemente considerada para se evitar recursos duplicados e não desperdiçar tempo e trabalho de profissionais em uma atividade desnecessária.
<b>C: Clientes podem customizar a tela de acordo com suas necessidades * *</b>	O sistema em questão é utilizado por vários clientes, logo ter a possibilidade de adequá-lo a sua realidade torna-o mais amigável e atende de forma mais completa as necessidades dos clientes.
<b>C: Maior flexibilidade, o usuário não precisa fazer a alteração em um sistema separado * *</b>	É importante que o cliente tenha uma experiência única na utilização do sistema e os subsistemas que compõem o todo devem ser transparentes ao usuário final.
<b>C: Reutilização de recursos * *</b>	A reutilização de recursos existentes, desde que adequada à realidade do projeto, deve ser fortemente considerada para se evitar recursos duplicados e não desperdiçar tempo e trabalho de profissionais em uma atividade desnecessária.
<b>Crítérios de relevância média</b>	
<b>Critério</b>	<b>Descrição</b>
<b>C: Recurso extra a implementar*</b>	Trabalho extra, cujo benefício real precisa ser bem avaliado.
<b>C: Funcionalidade extra e não estritamente necessária *</b>	Implica em trabalho extra e é necessário avaliar a real vantagem de implementá-la.
<b>C: Afeta o cronograma disponível *</b>	É necessário avaliar se o impacto no cronograma é justificável.

## C.4 Objeto do Experimento

O sistema utilizado como base para este estudo de caso é um gerenciador de projetos de pesquisa acadêmicos. O objetivo do sistema é possibilitar a gerência de um projeto de pesquisa, desde seus recursos humanos até seus recursos financeiros e operacionais. A figura a seguir exibe as principais metas para uso do sistema, as quais serão descritas, em maior detalhe, na seqüência.

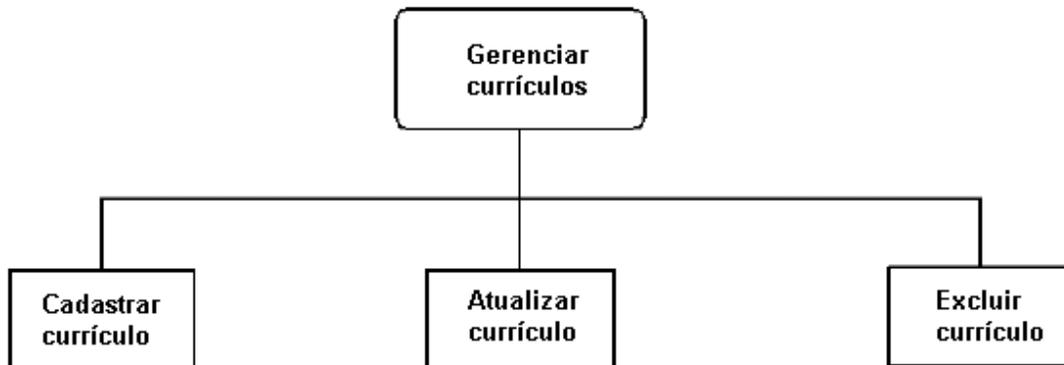


As três grandes metas de uso do Gerenciador (vistas na figura anterior) são: Gerenciar RH, Gerenciar Recursos e Gerenciar Projetos, as quais são descritas em detalhe a seguir:

- **Gerenciar RH:** necessidade de gerenciamento dos recursos humanos envolvidos no projeto. É subdividida em:
  - **Contratar e desligar funcionário:** procedimentos de contratação e desligamento de colaboradores do projeto;
  - **Gerenciar currículos:** gerenciamento da base de currículos (cadastramento, atualização e exclusão de currículos) de colaboradores ou de possíveis colaboradores do projeto;

- **Controlar prazos e contratos:** gerenciamento dos contratos com os colaboradores e de seus prazos (inclusão, renovação e consulta de contratos);
- **Controlar prazos e frequências:** gerenciamento da grade de horários de todos os colaboradores do projeto, ou seja, quais são as atividades dos colaboradores e os horários em que os mesmos estão envolvidos com estas atividades. Além disso, deve ser permitido o gerenciamento do ponto dos colaboradores, para verificar se os mesmos estão cumprindo com carga horária necessária ao projeto.
- **Gerenciar recursos:** necessidade de gerenciamento dos recursos materiais e financeiros do projeto. É subdividida em:
  - **Gerenciar fomentos:** responsável pelo gerenciamento dos recursos financeiros disponíveis para um projeto. Através desta funcionalidade é possível verificar os recursos alocados e disponíveis para serem alocados no projeto;
  - **Gerenciar ativo imobilizado:** funcionalidade que permite o gerenciamento dos recursos materiais envolvidos no projeto, como móveis, equipamentos, computadores, etc.
- **Gerenciar projetos:** necessidade de gerenciamento das atividades do projeto, bem como resultados, eventos e atividades externas que digam respeito ao projeto. É subdividida em:
  - **Gerenciar publicações:** gerenciamento das publicações geradas a partir dos resultados obtidos com o projeto;
  - **Gerenciar *deadlines*:** gerenciamento dos prazos tanto do projeto quanto de eventos que sejam de interesse do mesmo.

O escopo deste estudo de caso está restrito a uma parte do gerenciador de projetos. A parte escolhida para ser analisada é o **gerenciador de currículos**, que faz parte do módulo de gerência de RH. Basicamente, este gerenciador consiste nas operações básicas de leitura, gravação e exclusão de currículos, como pode ser visto na figura abaixo.



## C.5 Roteiro para Modelagem

O sistema anteriormente descrito deverá ser utilizado como objeto da modelagem a ser executada.

Para o desenvolvimento da modelagem deverão ser seguidos os passos abaixo:

1. Ler a descrição do sistema apresentado neste documento;
2. Fazer a modelagem do gerenciador de currículos, utilizando a técnica apresentada. Para isso:
  - a. Identificar as principais questões relacionadas ao projeto da funcionalidade e classificá-los quanto a sua importância;
  - b. Definir opções para resolver estas questões;
  - c. Definir os critérios a favor ou contra cada opção e classificá-los quanto a sua importância.
3. Após a modelagem diagramática, utilizar as tabelas apresentadas no modelo conceitual da técnica, para descrever sucintamente a importância das questões e critérios destacados.

Observação: a modelagem pode ser feita à mão livre, não necessitando uma ferramenta computacional para sua construção.

## C.6 Resultado esperado

Através do desenvolvimento desta modelagem espera-se obter como resultado o *feedback* sobre a aplicabilidade da técnica, nível de dificuldade para entendimento e aplicação da mesma, identificação de problemas e sugestões de melhorias. Para nos ajudar neste ponto, pedimos que o seguinte questionário seja respondido:

1. Quanto tempo você levou para realizar a modelagem.

2. Como você classificaria esta técnica em termos de dificuldade para os itens abaixo?

- a. Entendimento

- b. Aplicação

3. Você consegue visualizar a aplicação da mesma nos projetos nos quais trabalha? Se sim, em que fase do projeto?

4. Você acredita que esta técnica agregaria benefícios para projetos de software? Se sim, poderia citar algum?

5. Você acredita que as informações/experiências capturadas pela utilização desta técnica podem ser reaproveitadas em outros projetos?

## APÊNDICE D

### Termo de Consentimento Livre e Esclarecido

A equipe do **Laboratório de Usabilidade e Acessibilidade** agradece, a todos os participantes de experimentos realizados sob sua responsabilidade, a inestimável contribuição que prestam para o avanço das pesquisas na área de Interação Humano-Computador.

\*\*\*

O objetivo deste trabalho é investigar questões relacionadas à documentação de projeto de sistemas interativos. Para isto, os participantes do experimento são convidados a analisar as possibilidades de uso de um método proposto para a documentação das decisões de projeto de software. Esta análise será registrada em papel e estas informações nos trarão dados importantíssimos para dar continuidade a nosso trabalho de pesquisa.

O uso que se faz dos registros efetuados durante a entrevista é **estritamente** limitado a atividades de pesquisa, garantindo-se para tanto que:

1. O anonimato dos participantes será preservado em todo e qualquer documento divulgado em foros científicos (tais como conferências, periódicos, livros e assemelhados) ou pedagógicos (tais como apostilas de cursos, *slides* de apresentações, e assemelhados).
2. Todo participante terá acesso a cópias destes documentos após a publicação dos mesmos.

<p>Por favor, indique sua posição em relação aos termos acima:</p> <p><input type="checkbox"/> Estou de pleno acordo com os termos acima.</p> <p><input type="checkbox"/> Ao lado registro condições adicionais para este experimento.</p> <p>_____</p> <p style="text-align: center;">Assinatura do participante</p>	<p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p> <p><input type="checkbox"/> continua no verso</p>
---	--

**Nome do Participante:** \_\_\_\_\_

**Pesquisador Responsável:** Profa. Milene Selbach Silveira – Faculdade de Informática - PUCRS