



FACULDADE · DE · CIÊNCIAS UNIVERSIDADE · DE · LISBOA

NEGOCIAÇÃO DE REQUISITOS NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

João Jorge da Costa Vieira Ramires

Dissertação submetida para obtenção do grau de
MESTRE EM INFORMÁTICA

Orientador:

Pedro Alexandre de Mourão Antunes

Júri:

José Creissac Campos

Ana Luísa Correia Respício

Ana Paula Pereira Afonso

Novembro de 2004

NEGOCIAÇÃO DE REQUISITOS NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

João Jorge da Costa Vieira Ramires

Dissertação submetida para obtenção do grau de
MESTRE EM INFORMÁTICA

pela

Faculdade de Ciências da Universidade de Lisboa

Departamento de Informática

Orientador:

Pedro Alexandre de Mourão Antunes

Júri:

José Creissac Campos

Ana Luísa Correia Respício

Ana Paula Pereira Afonso

Novembro de 2004

Resumo

Estudos recentes comprovam que dos projectos de software concluídos, apenas uma pequena parte corresponde às expectativas, tendo-se observado que o problema se centra principalmente numa deficiente análise de requisitos.

Nas fases iniciais do processo de desenvolvimento de software, os intervenientes do processo contribuem com ideias vagas e incompletas relativamente aos seus objectivos. É frequente não haver uma ideia clara de quais são os requisitos desejáveis. Deste modo é difícil definir requisitos a fim de se obter um sistema que corresponda às expectativas.

Por outro lado, a avaliação de requisitos de software consiste no confronto dos objectivos dos intervenientes perante um conjunto de constrangimentos. Os conflitos nesta avaliação resultam de perspectivas, interpretações e posições diversas dos intervenientes.

Esta dissertação apresenta uma abordagem para a resolução do conflito na avaliação de requisitos de software. A abordagem proposta apresenta um sistema que interliga um grupo de pessoas – utilizadores, programadores, chefes de projecto, gestores de topo, etc. e suporta técnicas de negociação na resolução dos conflitos que surgem na avaliação dos requisitos de software.

O protótipo desenvolvido é simultaneamente um Sistema de Suporte à Decisão em Grupo e um Sistema de Suporte à Negociação assente no modelo Desdobramento da Função Qualidade aplicado ao software, fomentando uma perspectiva de negociação integrativa.

PALAVRAS-CHAVE: Sistemas de Suporte à Decisão em Grupo (GDSS), Sistemas de Suporte à Negociação (NSS), Desdobramento da Função Qualidade aplicado ao Software (SQFD).

Abstract

Recent studies prove that of the concluded software projects, only one small part corresponds to the expectations, having themselves observed that the problem centers mainly in a deficient requirements analysis.

In the initial phases of the software development process, the intervening ones of the process contribute with vacant and incomplete ideas relatively to its goals. It is frequent not to have a clear idea of which are the desirable requirements. In this way it is difficult to define requirements in order to get a system that corresponds to the expectations.

On the other hand, the evaluation of software requirements consists in the confrontation of the goals of the intervening ones against a set of constraints. The conflicts in this evaluation result of diverse perspectives, interpretations and positions of the intervening ones.

This dissertation presents a boarding for the resolution of the conflict in the evaluation of software requirements. The boarding proposal presents a system that establishes connection between a group of people - users, programmers, heads of project, top managers, etc. and supports negotiation techniques in the resolution of the conflicts that appear in the evaluation of software requirements.

The developed prototype is simultaneously a Group Decision Support System and a Negotiation Support System seats in the Software Quality Function Deployment model, fomenting a perspective of integrative negotiation.

KEY-WORDS: Group Decision Support System (GDSS), Negotiation Support System (NSS), Software Quality Function Deployment (SQFD).

Agradecimentos

Ao meu orientador, Professor Pedro Antunes, que pelo incentivo, crítica construtiva e constante disponibilidade, permitiram a concretização do trabalho apresentado.

Aos colegas e responsáveis da Unidade de Informática do Centro Nacional de Pensões, pelas críticas e sugestões.

Aos meus pais, pelo apoio que sempre me deram.

À minha filha e mulher, a quem não dei mais tempo como gostaria.

A Deus, pela sabedoria.

Lisboa, Novembro de 2004

João Jorge da Costa Vieira Ramires

Índice

Índice i

Lista de figuras vi

Lista de tabelas ix

1. Introdução 1

1.1 Contexto 1

1.2 Introdução aos temas abordados 2

1.2.1 Negociação de requisitos 2

1.2.2 Metodologia direccionada para a Qualidade 4

1.2.3 Suporte à argumentação 7

1.2.4 Consenso e Voto Maioritário 9

1.3 Abordagem proposta 10

1.4 Estrutura da tese 13

2. Negociação de Requisitos 14

2.1 Negociação 14

2.1.1 Estratégia integrativa 15

2.1.2 Estratégia distributiva 17

- 2.1.3 Dilema estratégico 18
- 2.1.4 Estilos individuais de gestão do conflito 21
 - Competição 21
 - Acomodação 21
 - Compromisso 22
 - Colaboração 23
 - Inacção 23

2.2 Requisitos de software 25

- 2.2.1 Engenharia de requisitos 25
 - Falsos requisitos 26
 - Elaboração de requisitos 27
 - Requisitos funcionais e não funcionais 28
- 2.2.2 Processo de engenharia de requisitos 29
 - Problemas no processo de engenharia de requisitos 30
 - Priorização de requisitos 31
 - Avaliação dos riscos 32
 - Listas para análise de requisitos 33
 - Princípios fundamentais da engenharia de requisitos 34

2.3 Negociação de Requisitos 34

- 2.3.1 Análise de requisitos e negociação 35
- 2.3.2 Planeamento dos conflitos e resolução 35
- 2.3.3 Reuniões de negociação 36

2.4 Conclusões 37

3. Modelos 39

3.1 Contexto 39

3.2 Desdobramento da Função Qualidade 41

3.3 Desdobramento da Função Qualidade no Software 49

3.4 IBIS 53

3.5 Ontologia dos argumentos 55

3.6 Consenso e Voto Maioritário 57

3.7 Conclusões 60

4. Trabalhos relacionados 62

4.1 Contexto 62

4.2 Suporte básico dos sistemas de negociação 63

4.2.1 Sistemas de correio electrónico 63

4.2.2 Placares informativos 63

4.2.3 Bases de dados partilhadas 63

4.3 Tipologias de sistemas de suporte à negociação 64

4.3.1 Sistemas de Suporte à Negociação (NSS) 64

4.3.2 Sistemas de Suporte à Decisão em Grupo (GDSS) 65

4.3.3 Comparação entre NSS e GDSS 66

4.4 Sistemas de suporte ao processo de negociação 67

4.4.1 EasyWinWin 67

4.4.2 MEDIATOR 69

4.4.3 HERMES 69

4.4.4 VIRTUAL QFD 70

4.4.5 CO-DECIDE 72

4.4.6 Pontos considerados no protótipo a desenvolver 72

4.5 Conclusões 75

5. Solução proposta 77

5.1 Contexto 77

5.2 Negociação no modelo SQFD 77

5.3 Máquina de estados 83

5.4 Processo de negociação 92

5.4.1 Fase de pré-negociação 92

5.4.2 Fase de negociação 94

5.4.3 Fase de pós-negociação 95

5.5 Esboço do protótipo 96

5.7 Conclusões 100

6. Protótipo desenvolvido 102

6.1 Contexto 102

6.2 Arquitectura 103

6.2.1 Microsoft EXCEL 104

6.2.2 MEGCLIENT 107

6.2.3 MEGSERVER 117

6.2.4 RTDSERVER 140

6.2.5 Repositório de Dados 141

7. Avaliação preliminar 146

7.1 Contexto 146

7.2 Experiência 1 147

7.2.1 Dados recolhidos 149

7.3 Experiência 2 156

7.3.1 Dados recolhidos 158

7.4 Análise de resultados 158

7.5 Conclusões 163

8. Conclusões e trabalho futuro 165

Referências 171

Lista de figuras

1.1 Esquema simplificado de uma matriz QFD 5

1.2 Sequência de matrizes QFD 6

1.3 Desdobramento de uma característica da matriz HoQ em matriz de sub-características 7

1.4 Sistema MEG no âmbito da Gestão de Qualidade Total (TQM) 10

1.5 Conflito entre stakeholders para um valor de correlação 11

2.1 Atitude dos stakeholders em situação de conflito 22

3.1 QFD-estendido em quatro fases 43

3.2 Esquema simplificado da obtenção de requisitos e soluções da Casa da Qualidade 45

3.3 Esquema das principais áreas da Casa da Qualidade 47

3.4 Cálculo de soluções prioritárias 48

3.5 QFD aplicado ao Software 52

3.6 Exemplo de um requisito do Cliente no SQFD 53

3.7 Esquema simplificado do modelo IBIS e correspondência com a Casa da Qualidade do modelo QFD 54

3.8 Esquema de suporte à negociação de correlações na Casa da Qualidade	61
5.1 Conflito entre stakeholders para um valor de correlação	78
5.2 Atitudes distributivas e integrativas	80
5.3 Máquina de Estados	84
5.4 Processo de negociação	93
5.5 Esquema de troca de informação	94
5.6 Esquema de solução proposta	95
5.7 Esboço da interface da <i>fase de pré-negociação</i>	97
5.8 Esboço da interface da <i>fase de negociação</i>	98
5.9 Esboço da interface da <i>fase de pós-negociação</i>	99
6.1 Arquitectura do sistema MEG	103
6.2 Exemplo de Casa da Qualidade para Software representada no sistema MEG	106
6.3 Formulário – <i>fase de pré-negociação</i>	107
6.4 Formulário – <i>fase de negociação</i>	109
6.5 Relação entre o formulário da figura 6.4 e a figura 3.8	110
6.6 Formulário VERSÃO TESTE – <i>fase de pré-negociação</i>	112
6.7 Formulário – <i>fase de pós-negociação</i>	113
6.8 Mensagem de <i>fase de pós-negociação</i> com <u>I'll agree</u>	114
6.9 Mensagem em caso de consenso	114

- 6.10 Mensagem em caso de falha de consenso 115**
- 6.11 Formulário – fase de pós-negociação com FIRM 115**
- 6.12 Formulário – fase de negociação com LOCK 117**
- 6.13 Formulário – MEGSERVER 118**
- 7.1 Casa da Qualidade para Software (projecto Cálculo de Prestações)
representada no sistema MEG 154**
- 7.2 Formulário – fase de negociação 155**
- 8.1 Modelo SQFD com suporte à negociação de correlações 167**

Lista de tabelas

1.1 Alguns pontos comparativos entre consenso e voto maioritário 9

3.1 Definições importantes no SQFD 50

3.2 Pontos comparativos entre consenso e voto maioritário 58

**4.1 Resumo dos principais aspectos que influenciaram o desenho do sistema
MEG 75**

7.1 Recursos utilizados na experiência no CNP 156

Capítulo 1

Introdução

1.1. Contexto

Estudos recentes comprovam que dos projectos de software concluídos, apenas uma pequena parte corresponde às expectativas, tendo-se observado que o problema se centra principalmente numa deficiente análise de requisitos [Grunbacher & Briggs, 2001].

Nas fases iniciais do processo de desenvolvimento de software, os intervenientes do processo contribuem com ideias vagas e incompletas relativamente aos seus objectivos. É frequente não haver uma ideia clara de quais são os requisitos desejáveis. Deste modo é difícil definir requisitos a fim de se obter um sistema que corresponda às expectativas. Por outro lado, é normal que exista conflito de interesses entre os vários requisitos propostos pelos diversos intervenientes.

Pretende-se apresentar nesta dissertação uma solução composta por um sistema colaborativo que auxilie as técnicas de negociação na resolução de conflitos durante a avaliação de requisitos de software, de modo a garantir a qualidade do produto através de um processo de comunicação que interligue um grupo de pessoas – clientes, utilizadores, programadores, chefes de projecto, gestores de topo, etc. normalmente denominados por *stakeholders*.

1.2. Introdução aos temas abordados

São definidos de seguida conceitos fundamentais para a percepção do trabalho descrito.

1.2.1 Negociação de requisitos

Os *requisitos* são definidos durante as fases iniciais do desenvolvimento do sistema como uma especificação do que terá de ser implementado. Tratam-se de *descrições de como o sistema se deve comportar ou uma propriedade / atributo do sistema*, o que significa que devem ser constrangimentos no processo de desenvolvimento [Sommerville & Sawyer, 1997]. Por outro lado, Lewicki & Litterer (1985) definem *negociação* como *o processo social básico usado para resolver conflitos*. Um requisito que pode ser aceite por um stakeholder pode não ser aceite por outro. O processo de decisão na negociação envolve procurar uma única alternativa (ponto de intersecção) dentro de um conjunto de alternativas aceitáveis pelos stakeholders e tecnologicamente possíveis. Durante uma negociação, estes conjuntos podem sofrer alterações, expandindo ou contraindo até ser encontrado um ponto de intersecção [Kersten, 1987].

A negociação de requisitos é uma das fases iniciais de qualquer ciclo de desenvolvimento de software, mas o seu impacto reflecte-se a nível global do sistema. Depois de obtermos um conjunto de requisitos, deve ser efectuada uma análise cuidada de forma a procurar conflitos, sobreposições, omissões e inconsistências. Após esta etapa, os stakeholders negociam a aceitação de um conjunto de requisitos. Os conflitos devem ser resolvidos e os requisitos ordenados pela sua importância [Sommerville & Sawyer, 1997]. A recolha, análise e negociação de requisitos estão intimamente ligados e fazem parte de um processo em espiral. Este processo deve ser seguido até à satisfação de todas as partes envolvidas, sendo um processo influenciado não só pela lógica e argumentos técnicos mas também pela política da organização e pela personalidade dos stakeholders [Sommerville & Sawyer, 1997].

A negociação actuará a dois níveis [Belluci & Zleznikov, 1998]:

- individual, onde cada stakeholder terá uma noção mais clara dos seus próprios objectivos e metas;
- colectivo, em que se negociarão os objectivos e metas divergentes para dar lugar a um novo objectivo global. Trata-se portanto de um processo gradual de aceitação individual conduzido pelas restrições expostas pelo grupo, onde as expectativas do sistema a construir correspondem à realidade possível – visão conjunta do projecto que compromete todos os stakeholders.

Um sistema colaborativo com estas características permite o acesso ao conhecimento tático de forma diferente dos métodos tradicionais, como os baseados em reuniões ou mesmo em entrevistas. Por outro lado, permite juntar de forma eficiente todos os stakeholders focando a atenção em interesses bem definidos, evitando a divagação no diálogo e no conflito [Ramesh & Bui, 1999] como acontece no caso das reuniões, ou evitar as dificuldades de integração, como no caso das entrevistas.

O trabalho proposto baseia-se no modelo de desenvolvimento em espiral (processo iterativo e análise de risco) em que a compreensão / resolução de conflitos se revela uma estratégia importante na diminuição do risco [Bohem et al.,1995] [Robinson & Volkov, 1998]. Pretende-se que a ferramenta a desenvolver:

- clarifique os objectivos concretos dos stakeholders;
- faça emergir a experiência e o conhecimento tático dos stakeholders;
- explicitar os diferentes pontos de vista dos stakeholders e permitir negociar os conflitos através da apresentação de alternativas;
- permitir uma melhor visualização da memória de grupo;

- focalizar a comunicação.

1.2.2 Metodologia direcionada para a Qualidade

Apesar das metodologias tradicionais de desenvolvimento de sistemas reconhecerem a importância dos stakeholders, elas não fornecem métodos formais que transformem os requisitos de qualidade desses stakeholders em especificações de desenho do sistema. Por outro lado, as metodologias não são rigorosas a manter a atenção nos requisitos de qualidade definidos pelos stakeholders nas *fases posteriores* de desenvolvimento [Stylianou et al., 1997].

Uma das técnicas utilizadas em organizações que implementam programas de *Gestão de Qualidade Total* (Total Quality Management, TQM) [Zultner, 1993] [Stylianou et al., 1997] para manter os requisitos de qualidade dos stakeholders por todo o processo de desenho, produção e entrega dos produtos denomina-se *Desdobramento da Função Qualidade* (Quality Function Deployment, QFD) [Zultner, 1993], tendo sido aplicada mais tarde ao processo de desenvolvimento de software (Software Quality Function Deployment, SQFD) [Haag et al., 1996].

O QFD pode ser definido como *um desdobramento passo-a-passo de uma função ou operação que implica qualidade, em detalhes através da sistematização de alvos e significados* [Akao, 1997]. O QFD foi concebido no Japão pelo Dr. Akao em meados do anos 60, numa época em que as indústrias japonesas passaram da imitação de produtos para o desenvolvimento baseado na originalidade. No domínio do software, a qualidade exigida é cada vez maior, sendo que através do QFD se consegue [Akao, 1997]:

- identificar essa qualidade;
- um método para a construção de sistemas que assegurem essa qualidade.

A metodologia QFD distingue entre requisitos e soluções de implementação, podendo os seus objectivos ser resumidos em [Stylianou et al., 1997]:

- identificação dos stakeholders que pretendem o sistema;
- identificação do *que* desejam;
- identificação de *como* satisfazer esses desejos.

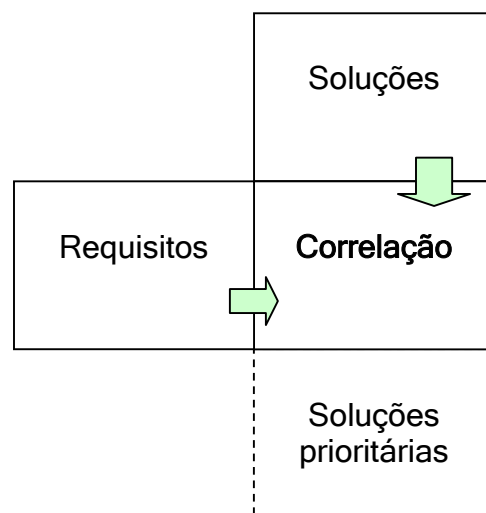


Figura 1.1. Esquema simplificado de uma matriz QFD

No QFD são usadas várias matrizes em que se comparam os requisitos com as soluções na forma de correlações. Para cada uma destas correlações são possíveis os valores $\{ 0, 1, 3, 9 \}$ que identificam a *medida* da correspondência entre a solução apresentada e o requisito proposto (figura 1.1). Neste modelo a primeira matriz assume um papel preponderante, denominada *Casa da Qualidade* (House of Quality, HoQ), onde são relacionados os requisitos dos stakeholders que pretendem o sistema com as soluções propostas pelos stakeholders que procedem ao seu desenvolvimento – procura-se anular

a hipótese de um não entendimento das necessidades reais dos primeiros por parte dos segundos. A partir das correlações definidas são obtidos pesos para cada solução, permitindo ordenar pela sua importância (priorizar) as soluções da matriz.

É de salientar que a HoQ permite igualmente o confronto com os produtos da concorrência: avalia-se o posicionamento do produto face a outros produtos, permitindo otimizações e soluções originais que acrescentem valor.

O QFD promove o desdobramento dos requisitos / soluções em múltiplas matrizes que encaixam umas nas outras num processo semelhante ao refinamento das metodologias tradicionais (figura 1.2).

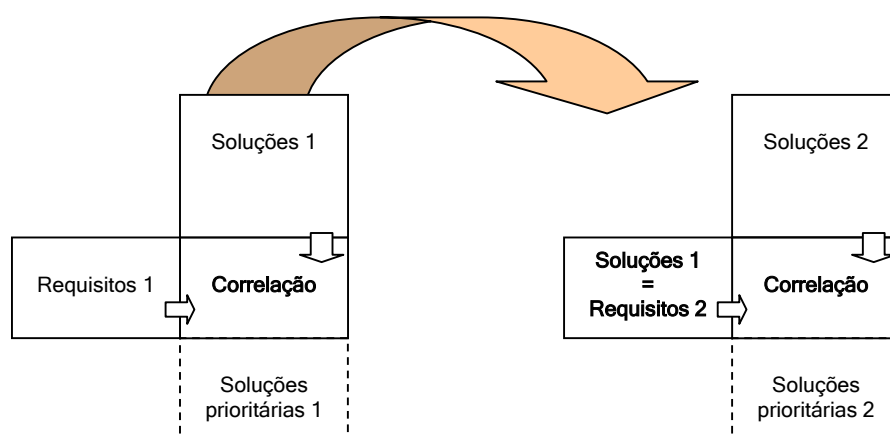


Figura 1.2. Sequência de matrizes QFD

Um exemplo de requisito para Requisitos 1 descrito em linguagem natural: *curto tempo de resposta*, pode ser correlacionado com um termo técnico em Soluções 1: *velocidade de hardware*. Por sua vez, *velocidade de hardware* (agora em Requisitos 2) pode ser correlacionado numa segunda matriz com os termos *uso de componentes rápidos*, *tolerância a falhas* e *recuperação de erros* presentes em Soluções 2 (figura 1.3).

O princípio da distinção entre requisitos e soluções pode ser vista em todas as etapas do processo de desenvolvimento dado que os resultados de uma etapa podem ser interpretados como sendo os requisitos de outra etapa e assim sucessivamente.

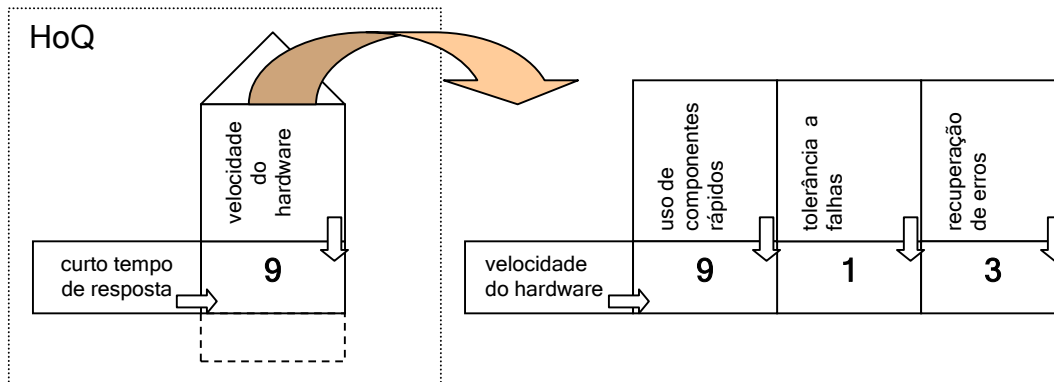


Figura 1.3. Desdobramento de uma característica da matriz HoQ em matriz de sub-características

O presente trabalho focaliza-se na primeira matriz do SQFD (software-HoQ), sendo a solução igualmente válida para as outras matrizes.

1.2.3 Suporte à argumentação

A teoria que sustenta os modelos de negociação normalmente usados promove por um lado a identificação dos objectivos mais importantes de cada stakeholder e por outro a garantia de que esses objectivos são atingidos [Nuseibeh & Easterbrook, 2000].

No QFD são usadas matrizes em que se relacionam requisitos / soluções sem mostrar explicitamente os objectivos dos stakeholders. Importa por isso expor as razões que sustentam uma posição de um stakeholder face a uma alternativa proposta por outro

stakeholder no processo de negociação. Uma solução será proporcionar suporte à argumentação.

O sistema de suporte à argumentação mais popular, IBIS [Conklin & Begeman, 1988] utiliza *questões, posições e argumentos* para mediar a conversação entre stakeholders. Basicamente, no modelo IBIS existe uma *questão* que serve de base à discussão, onde as *posições* dos vários stakeholders reflectem as opiniões a favor ou contra a *questão* e os *argumentos* suportam as posições. O modelo IBIS pode ser usado para representar um processo de discussão através do QFD [Jacobs & Kethers, 1994].

O suporte à argumentação visa aumentar a racionalidade do processo de negociação, aumentando inclusive a memória organizacional pelo registo dos argumentos que levam a uma determinada posição relativamente a uma questão que dá origem ao processo. No entanto, o suporte à argumentação restringe-se a uma iteração do processo de negociação que começa quando existe conflito e termina quando o conflito é resolvido. Para um novo processo de negociação, se for pretendido justificar as posições assumidas, torna-se necessário inserir novamente os argumentos.

Na abordagem proposta, os argumentos possíveis são restringidos a um vocabulário de termos de modo a reduzir a dispersão dos argumentos (não é permitida a escrita de argumentos). O vocabulário de termos que serve de base aos argumentos consiste num *conjunto de palavras reservadas que fazem sentido num determinado contexto* – ontologia [Lin et al., 1996].

Pretende-se que o conhecimento tático e a experiência dos stakeholders seja revelada. Baseado numa ontologia, o suporte à argumentação permite focalizar a comunicação do grupo em itens específicos evitando a verbalização.

1.2.4 Consenso e voto maioritário

Um processo de negociação pode ser resolvido por:

- consenso, onde é necessário a aceitação de uma *questão* por todas as partes envolvidas;
- voto maioritário, onde a vontade do maior grupo prevalece.

Tabela 1.1 Alguns pontos comparativos entre consenso e voto maioritário

Consenso	Voto maioritário
Promove cooperação	Promove competição
Todos os stakeholders têm o mesmo poder de decisão	O poder de decisão corresponde a hierarquia, popularidade, etc.
Os stakeholders concordam ou não concordam	Os stakeholders votam contra, a favor ou abstêm-se
Todas as ideias são ouvidas e exploradas	A maioria decide quais as ideias a serem consideradas

Uma análise à tabela 1.1 sugere que para uma negociação ser bem sucedida, a estratégia a seguir deve ser baseada em consenso. No entanto, o voto maioritário pode ajudar a medir a aproximação a um consenso, significando que pode ser considerado um mecanismo de ajuda à negociação.

No trabalho proposto tem-se que a solução apresentada fornece suporte à resolução do conflito por consenso na avaliação de requisitos de software fornecendo simultaneamente informação quanto à preferência da maioria dos stakeholders.

1.3. Abordagem proposta

Este trabalho propõe um sistema para a resolução de conflitos na avaliação de requisitos de software. O protótipo desenvolvido (sistema MEG) visa interligar os stakeholders e proporcionar um meio simples de negociação dos requisitos de software.

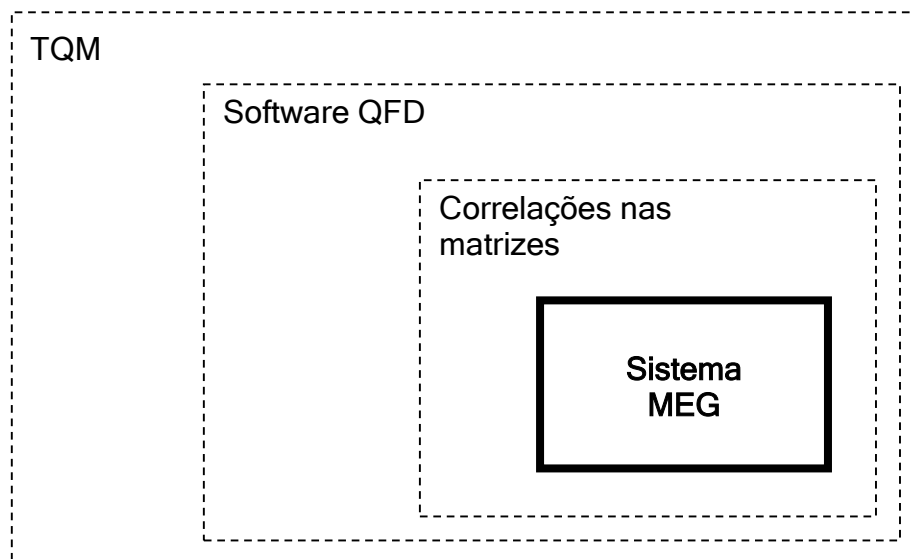


Figura 1.4. Sistema MEG no âmbito da Gestão de Qualidade Total (TQM)

O sistema MEG suporta a negociação de requisitos ao nível da matriz software-HoQ (figura 1.4). Os diferentes stakeholders podem ter vários valores possíveis para as correlações requisitos / soluções da matriz, o que pode originar conflito de interesses, exigindo por isso negociação. A resolução do conflito entre valores das correlações afecta directamente a priorização dos requisitos no SQFD. Pretende-se portanto um

sistema que auxilie a resolução de conflitos entre as correlações definidas pelos stakeholders. A figura 1.5 apresenta uma situação de conflito para um valor de correlação c entre dois stakeholders S_i e S_j .

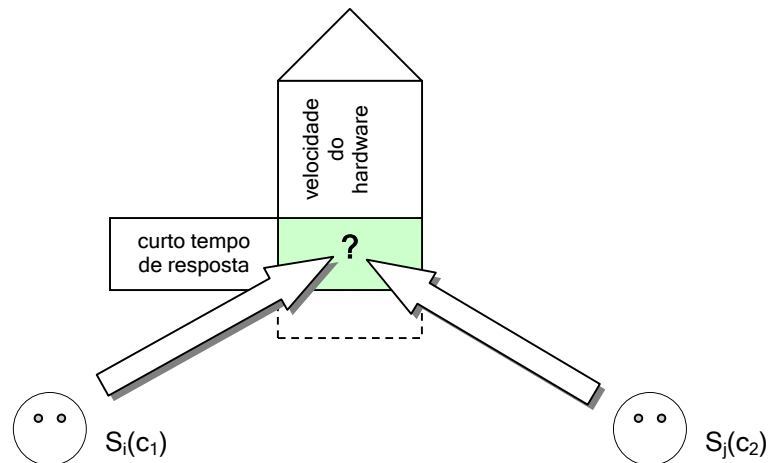


Figura 1.5. Conflito entre stakeholders para um valor de correlação

Para representar o processo de argumentação na negociação de conflitos recorre-se a um modelo de argumentação do tipo IBIS: uma *questão* será o primeiro valor de correlação proposto na matriz HoQ e as *posições* são reflectidas a partir dos outros valores propostos pelos stakeholders – se o valor inserido é igual, será uma *posição* a favor da *questão* e contra caso contrário. As *posições* podem ser apoiadas por *argumentos*.

O modelo de argumentação utilizado no sistema MEG reflecte de forma *automática* as posições tomadas pelos stakeholders face às questões que geram conflito. O processo de negociação é apresentado num formato de *fórum de discussão*, onde são mostradas as posições sem contudo divulgar os valores que as originaram, nem os valores que os stakeholders estão dispostos a aceitar, mantendo a reserva necessária a um processo de negociação (apenas o sistema conhece os valores de todos os stakeholders).

As diferentes atitudes que os stakeholders podem tomar face ao conflito durante o processo de negociação [Lewicki & Litterer, 1985] – *competição, acomodação, compromisso, colaboração* ou *inacção*, são encapsuladas no sistema MEG de modo a esconder a complexidade do processo de negociação.

O sistema poderá propor um valor obtido a partir de voto maioritário dos valores propostos pelos stakeholders. Apesar de o sistema propor um valor, não impõe que este seja o escolhido – apenas tenta mostrar a situação global e orientar na decisão mais justa (mecanismo de ajuda). A aceitação final de um valor é conseguida somente pela via negocial dos stakeholders presentes na discussão e resolvido por consenso.

Na abordagem proposta, cada stakeholder apenas tem de:

- escolher o(s) valor(es) de correlação que pretende;
- seleccionar os argumentos que suportam a sua posição quando necessário;
- aceitar ou não a solução apresentada pelo sistema.

Os stakeholders são confrontados com uma folha de cálculo Microsoft EXCEL que resume a software-HoQ situada no computador pessoal de cada stakeholder, apresentando em tempo real o resultado da inserção de valores das correlações dos outros stakeholders. Este processo é auxiliado por uma ferramenta de inserção e visualização do estado de negociação (MEGCLIENT) para cada célula escolhida da software-HoQ. A comunicação entre os vários clientes MEGCLIENT e a gestão do repositório de dados é conseguida por intermédio de um servidor (MEGSERVER), enquanto os programas EXCEL são sincronizados através de um outro servidor (RTDSERVER).

1.4. Estrutura da tese

No capítulo 2 são aprofundados os aspectos da negociação e negociação de requisitos de software.

No capítulo 3 é apresentado o modelo *Desdobramento da Função Qualidade* e sua aplicação ao software, suporte dado à argumentação, consenso e voto maioritário.

O capítulo 4 analisa trabalhos relacionados com os assuntos relatados.

O capítulo 5 descreve a solução proposta.

O capítulo 6 descreve o protótipo desenvolvido – sistema MEG.

O capítulo 7 apresenta uma avaliação preliminar do protótipo junto de stakeholders.

E por ultimo o capítulo 8 descreve as conclusões e trabalho futuro.

Capítulo 2

Negociação de Requisitos

2.1. Negociação

Lewicki & Litterer (1985) definem negociação como o *processo social básico usado para resolver conflitos*. As características que fazem da negociação o processo inevitável para a resolução de conflitos são [Lewicki & Litterer, 1985]:

- existe um conflito de interesses entre duas ou mais partes, onde o que um quer não é o que os outros querem;
- não existe um conjunto de regras, métodos ou autoridades que resolvam o conflito;
- não existe a possibilidade de experimentar varias soluções possíveis;
- o contrato não pode ser desfeito, implicando que tenha de ser estabelecido um compromisso para próprio interesse.

O processo de decisão na negociação envolve procurar uma única alternativa (ponto de intersecção) dentro de um conjunto de alternativas aceitáveis dos stakeholders e tecnologicamente possíveis. Na negociação, estes conjuntos podem sofrer alterações, expandindo ou contraindo até ser encontrado um ponto de intercepção [Kersten, 1987].

Ajenstat & Benchimol (1999) referem que a negociação é uma das aproximações do *Cooperative Distributed Problem Solving* (CDPS). O CDPS foi originalmente proposto como uma forma de estudar como é que um conjunto de entidades ligadas em rede conseguem trabalhar em conjunto para resolver problemas que individualmente não poderiam ser resolvidos. A estrutura do CDPS numa organização pode ser entendida como um conjunto de entidades em que cada entidade (nó) da rede pode trabalhar de forma independente na resolução de problemas complexos mas onde se necessita de cooperação para resolver os problemas a nível organizacional. Num projecto de desenvolvimento de software extenso, não existe um stakeholder suficientemente capaz nem possuidor de recursos, informação ou conhecimento suficientes para resolver um problema geral – cada stakeholder apenas tem capacidade para resolver parte da questão. Existem basicamente dois constrangimentos do processo:

- as soluções individuais devem ser compatíveis com a solução geral;
- a comunicação entre entidades deve ser feita de forma cooperativa.

Segundo Chaudhury et al. (1991) o processo de negociação pode ser dividido em duas estratégias: integrativa e distributiva (também denominadas na literatura respectivamente por simbiótica e predatória [Anderson, 1992]).

2.2.1 Estratégia integrativa

No processo integrativo tenta-se chegar a acordo de forma inventiva, cooperativa e de busca persistente de ganhos conjuntos e criação de valor, recorrendo-se à partilha de informação sem quaisquer reservas.

Neste processo podem acontecer duas situações:

- uma parte cede completamente aos objectivos da outra parte numa estratégia de submissão aberta, eventualmente com o fim de estabelecer um relacionamento positivo no futuro;
- uma estratégia onde ambas as partes ganham ou pelo menos nenhuma perde – negociação tipo win-win [Bohem et al., 2001]. Neste caso ambas as partes se esforçam por preservar um bom relacionamento futuro enquanto aumentam os ganhos substantivos da negociação para ambas as partes.

Em [Anderson, 1992] vêm descritos os critérios em que assenta a estratégia integrativa:

- a) separar as pessoas dos problemas: os intervenientes devem trabalhar lado a lado, atacando o problema em vez de se atacarem mutuamente;
- b) focar nos interesses, não em posições: as posições tomadas costumam obscurecer as necessidades reais, estando intimamente ligadas ao ego dos intervenientes;
- c) criar opções para ganho mútuo: devem ser criadas várias opções para se escolher qual o procedimento a efectuar;
- d) insistir no uso de critérios objectivos: o uso de modelos padronizados que promovam justiça (leis, valor de mercado, opinião de peritos) conduzem a discussão segundo um critério objectivo.

Anderson (1992) resume assim as estratégias integrativas:

- ênfase em canais de comunicação que não ocultem interesses, separando os aspectos substantivos das partes envolvidas;
- procura de acordos amigáveis através de relações saudáveis e duradouras;

- o foco nos interesses de cada um evita custos associados da troca de mensagens baseadas em posições pouco flexíveis, o que promove opções de mútua satisfação.

Numa estratégia integrativa procura-se suprir os mais legítimos interesses de cada parte, da melhor forma possível e com justiça.

2.1.2 Estratégia distributiva

Este tipo de estratégia é baseada numa posição dura e inflexível onde nenhum participante se preocupa com as necessidades dos outros. O objectivo é o de persuadir as outras partes de que eles querem o que nós temos para oferecer enquanto nós só raramente estamos interessados naquilo que os outros têm para nos oferecer. É um jogo de ganhar ou perder – negociação tipo win-lose [Bohem et al., 2001]. O resultado depende da habilidade para fazer concessões de forma lenta, exagerar no valor das concessões, minimizar o valor das concessões dos outros, ocultar informação, inquirir violentamente acerca de princípios vantajosos que implicam decisões favoráveis e apenas fazer acordos nessa situação. Fazem parte deste jogo ridicularizar, atacar ou intimidar e comportamentos intransigentes, retaliadores ou agressivos, assim como jogar com o tempo.

Anderson (1992) propõe diversos critérios a seguir por quem deseja optar por uma estratégia distribuída:

- a) negociar firme: não necessariamente de forma arrogante, rude ou implacável mas mostrar que se quer negociar e que não se tem receio de pedir concessões extras;

- b) examinar minuciosamente os detalhes: avaliar cada ponto e sempre que um item não esteja estabelecido tentar ganhar vantagem. Não assumir nada, perguntar por tudo e não considerar nenhum item insignificante;
- c) negociar arduamente cada item especialmente os menores – ver como reage a outra parte para se saber como fazer com pontos maiores;
- d) focar a atenção nos objectivos: saber diferenciar o que está a ser negociado do contexto envolvente, como o estado de espírito da outra parte, e saber contrariar a estratégia do adversário;
- e) evitar ultimatoss: proporcionar aos oponentes escolhas em vez de ultimatoss. O facto de não se dar saída ao oponente torna difícil prever a sua reacção;
- f) procurar pontos vulneráveis no oponente: por exemplo se uma fase do projecto se encontrar no limite do tempo estipulado, a aceitação de condições por parte de um opositor é mais fácil bastando para isso ocupá-lo no início com pormenores;
- g) controlar a negociação: fazer com que o opositor siga um plano que não o dele.

2.1.3 Dilema estratégico

As estratégias integrativas, desenhadas para criar valor através da cooperação e colaboração, entram em conflito com as distributivas, onde se pretende reclamar valor. Na realidade não interessa quão criativa ou colaborativa a negociação possa ser, nem que sucesso na criação de valor o processo origine, o que sucede é que a dada altura cada parte reclama uma parte do valor – as estratégias integrativas são vulneráveis às estratégias distributivas.

A negociação distributiva pretende reclamar valor, inibindo os esforços de satisfazer ambas as partes com a criação de valor. Maximizando o valor das concessões e

minimizando os benefícios cedidos pela outra parte, inibe-se o processo de comunicação. Intransigências não favorecem a escuta e entendimento dos interesses da outra parte. O uso excessivo da estratégia distributiva pode corromper as relações entre as partes, aumentando o conflito e diminuindo cada vez mais as possibilidades de acordo.

A negociação integrativa, ao disponibilizar a informação de forma clara para procurar o bem comum, torna-se vulnerável a estratégias que reclamam valor. Revelar os desejos genuínos de uma parte é arriscado e a boa vontade para se criarem novas ofertas pode ser interpretada como futuras concessões. Como resultado, geralmente essas ofertas que podem satisfazer interesses comuns não são reveladas. Mesmo em ambientes onde se pratique somente uma estratégia integrativa, os stakeholders podem ficar reféns da troca de concessões.

Mesmo que ambas as partes utilizem técnicas integrativas, devem estar alerta para o facto de algumas vezes terem de reclamar valor. Normalmente uma solução óptima surge da discussão aberta, respeitando os interesses e relacionamentos, procurando criar soluções que satisfaçam os interesses e focando os objectivos a longo prazo. No entanto, na realidade nem sempre esse comportamento acontece, dado que a parte que legitimamente emprega técnicas integrativas oculta factos que a tornam vulnerável a partes que sejam distributivas, originando que o clima de resolução de problemas seja substituído por um clima de competitividade que começa de forma distributiva, mesmo que de forma modesta. O uso da integração é condicionado pelo grau de desconfiança das técnicas usadas pelos outros. Esta suspeita origina que muitos ganhos conjuntos não sejam alcançados por permanecerem escondidos. Se ambos aplicarem estratégias distributivas a probabilidade de se criar valor é progressivamente eliminada, o que origina que ambas as partes obtenham resultados medíocres em termos de resultados substantivos e na relação entre si [Anderson, 1992].

Damian et al. (2003) salientam que os modelos de negociação de requisitos geralmente enfatizam a automatização da identificação e resolução de conflitos e prestam pouca atenção aos aspectos socio-psicológicos do processo de grupo. Estudos no campo mostram que o conflito entre diferentes grupos de stakeholders é uma séria ameaça para o sucesso dos projectos de software. Normalmente os stakeholders pertencem a unidades organizacionais que competem entre si. A estrutura de recompensas numa organização torna a negociação mais difícil, dado que os stakeholders são recompensados e promovidos com base no seu desempenho individual. Quando especificam requisitos para o sistema, podem actuar em proveito próprio de modo a aumentar o desempenho da sua unidade de negócio, mesmo que prejudicando as outras unidades ou a organização em geral – estratégia distributiva. Apesar de existirem oportunidades para acordos onde as partes podem maximizar ganhos conjuntos – estratégia integrativa, são conseguidos compromissos sub-óptimos em vez de procurar encontrar acordos que beneficiam todas as partes.

A compreensão do dilema estratégico melhora a escolha da melhor estratégia. Anderson (1992) defende que são conseguidos melhores resultados e relacionamentos se se optar por uma estratégia intermédia entre integrativa e distributiva:

- analisar cada item e consequências de concessões e até que ponto são negociáveis antes do processo de negociação – apesar de ser o menos interessante de fazer no processo de negociação, é o mais importante;
- negociar directamente com aqueles que têm autoridade para aceitar alterações;
- construir relações sempre que possível: mesmo usando tácticas distributivas, é sempre melhor negociar com um amigo do que com um inimigo;

- evitar concessões rápidas: avaliar as concessões mesmo que sejam pequenas, pois pode estar a ser cedido algo desnecessário ou que coloque em risco necessidades básicas;
- manter a postura: as emoções podem ficar descontroladas em certas situações. Procurar discernir os intentos da outra parte de forma racional.

2.1.4 Estilos individuais de gestão do conflito

A atitude que os stakeholders podem tomar no conflito e na negociação é essencial para se entender as diferenças entre uma estratégia integrativa e distributiva.

Lewicki & Litterer (1985) referem que existem fundamentalmente cinco atitudes que os stakeholders podem seguir numa situação de conflito: competição, acomodação, compromisso, colaboração e inacção. Esta classificação é baseada no grau de satisfação dos objectivos de cada stakeholder face ao grupo.

As atitudes representadas na figura 2.1 são descritas de seguida [Lewicki & Litterer, 1985].

Competição

Cada parte tenta convencer a outra parte a aceitar a posição que favorece apenas os seus próprios interesses (estratégia distributiva).

Acomodação

Em contraste à competição, uma estratégia de acomodação tenta ajudar a outra parte a obter os seus próprios objectivos. Em vez de um stakeholder se focar nas suas necessidades, foca-se nas necessidades dos outros. A primeira prioridade é assegurar que a outra parte está completamente satisfeita. Os stakeholders podem acomodar-se por

várias razões: para acabar a negociação, para deixar a outra parte completamente satisfeita agora para pedir algo mais tarde, ou porque as questões são significativamente mais importantes para a outra parte.

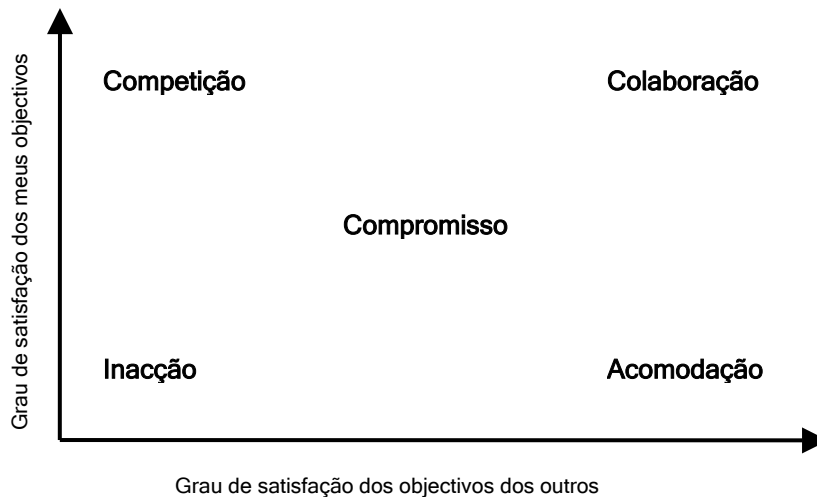


Figura 2.1. Atitude dos stakeholders em situação de conflito
[Lewicki & Litterer, 1985]

Compromisso

Quando os negociadores se comprometem, eles repartem os ganhos. Nem um nem outro ganha. Em vez disso, cada lado concorda nalguma divisão da questão de maneira que cada um obtém parte da solução – mas não tudo. As partes concordam no compromisso quando eles vêm a questão em termos de ganhos fixos e repartem as diferenças de modo a que cada um fique com uma parte. É um mecanismo para satisfação (assegurando que cada parte leva alguma coisa), mas não otimizado dado que cada parte não consegue alcançar todos os seus objectivos.

Colaboração

Quando os stakeholders colaboram, trabalham juntos para maximizar os resultados da sua união. As partes compreendem que partilham um problema e definem o problema em termos de alvos e interesses partilhados. Cada parte expõe as suas necessidades, e as partes trabalham juntas para inventar opções que preencham essas necessidades de uma maneira otimizada.

Inacção

As partes podem decidir evitar o conflito. Se um stakeholder nem puxa para si os seus próprios objectivos nem mostra preocupação pelos objectivos dos outros, ele está a evitar o conflito. Evitar é similar a retirar-se da situação de conflito. A consequência é que a outra parte pode fazer o que quiser ou, se a outra parte é dependente da primeira, esta retirada vai frustrar o opositor. Esta estratégia é uma aproximação que pode ser usada quando o stakeholder teme negociar, quando as questões são vistas como insignificantes para prosseguir a negociação ou quando se pretende protelar a negociação.

2.2. Requisitos de software

Os requisitos de software são definidos durante as fases iniciais do desenvolvimento do sistema como a especificação do que terá de ser implementado. Tratam-se de descrições de como o sistema se deve comportar ou uma propriedade / atributo do sistema, o que significa que devem ser constrangimentos no processo de desenvolvimento [Sommerville & Sawyer, 1997].

Um requisito pode descrever [Sommerville & Sawyer, 1997]:

- funções ao nível dos stakeholders que utilizam o sistema: *o processador de texto deve incluir um corrector automático;*
- uma propriedade genérica do sistema: *é necessário que qualquer operação seja antecedida de autenticação;*
- um constrangimento específico: *o sensor deve detectar qualquer movimento a cada t segundos;*
- um constrangimento no desenvolvimento do sistema: *o sistema deve ser desenvolvido em JAVA.*

Alguns autores como Tom Gilb (2003a) sugerem que os requisitos devem ser sempre algo que diga o *que* o sistema deve oferecer em vez de *como* fazê-lo. No entanto, na prática isso não se verifica [Sommerville & Sawyer, 1997]:

- os leitores do documento de requisitos são normalmente stakeholders que desenvolvem o sistema e que entendem muito mais facilmente uma descrição de implementação do que abstracções genéricas (os requisitos devem ser escritos para quem os lê);
- em muitos casos o sistema a implementar está integrado num contexto mais alargado de sistemas, pelo que se torna necessário especificar a implementação e a política da organização.

Os requisitos contém normalmente uma mistura de informação acerca do problema, comportamentos, propriedades e constrangimentos de desenho e implementação [Sommerville & Sawyer, 1997].

No modelo QFD exposto no capítulo 3, não só existe distinção entre os requisitos que são elaborados pelo stakeholders que pretendem o sistema e aqueles que são

responsáveis pelo seu desenvolvimento, como também existe separação entre requisitos e soluções em cada etapa do processo de desenvolvimento de software.

2.2.1 Engenharia de requisitos

A engenharia de requisitos abrange todas as actividades de recolha, documentação e gestão de requisitos na elaboração de um sistema. O uso do termo engenharia implica técnicas sistemáticas que assegurem requisitos completos, consistentes, relevantes, etc [Sommerville & Sawyer, 1997].

Gilb (2003a) refere engenharia como *o uso de princípios de forma sistemática, para encontrar o desenho do sistema que satisfaça os requisitos atendendo a um conjunto de constrangimentos, sob condições de incerteza* em que define:

- princípios: métodos práticos de fazer as coisas, que guiam as escolhas e acção dos stakeholders. Princípios não são regras fixas mas sim reflexo da experiência adquirida em projectos anteriores;
- forma sistemática: deve incluir mecanismos de geração de ideias e de avaliação do conhecimento adquirido;
- satisfação dos requisitos: só é conseguida na medida em que se consegue implementar todos os requisitos propostos, sendo um processo que se inicia no desenho do sistema até à sua implementação;
- condições de incerteza: as combinações de desenho são aplicadas num contexto em constante alteração. Os resultados são difíceis de prever e o risco tem de ser medido.

A definição de engenharia atrás descrita aplica-se à engenharia de software. Quanto à engenharia de requisitos, existe um caminho ainda longo a percorrer, existindo

inclusivamente equívocos do que são realmente requisitos, sendo facilmente confundidos com o que Gilb (2003a) chama de *falsos requisitos*.

Falsos requisitos

Falsos requisitos ocorrem quando uma especificação contém elementos que não são necessários ou desejáveis. Gilb (2003a) explana essas situações da forma seguinte:

- os requisitos não são colocados numa qualificação nem são quantificados: é comum encontrar requisitos como *custos reduzidos*, *promover a gestão de controlo*, *elevada usabilidade* e *aumentar a satisfação do cliente*, onde:
 - não existe uma clara definição dos objectivos que se pretendem;
 - não existe recurso a um conjunto de termos de base que reduzam ambiguidades;
 - não existe quantificação do nível de qualidade corrente nem do nível exigido de futuro.
- usar descrições de implementação como *é necessário uma base de dados orientada a objecto* ou *é preciso um interface Windows*, em vez de requisitos verdadeiros (não são especificados com clareza as razões que justificam a escolha das soluções apresentadas).

O uso de descrições de implementação na definição dos requisitos pode causar os seguintes efeitos negativos [Gilb, 2003a]:

- o requisito verdadeiro não é estabelecido / conhecido;
- melhores alternativas de implementação não são procuradas;

- o desenho de sistema não pode ser otimizado de forma a satisfazer o maior número de requisitos.

Os requisitos verdadeiros são melhor estabelecidos em termos de objectivos finais, em vez de como chegar a esses objectivos. Se se lidar com os requisitos a um nível de abstracção mais elevado, consegue-se uma maior diversidade de soluções de implementação.

Elaboração de requisitos

Cada requisito deverá ser escrito de forma a ser entendido pelo leitor alvo em cada etapa do processo de desenvolvimento de software. Muitos requisitos são especificados em linguagem natural suportados por tabelas / diagramas de informação detalhada.

O nível de detalhe depende da prática organizacional, do tipo de sistema e se o documento de requisitos vai ser a base do contrato do sistema a desenvolver. Algumas organizações especificam inicialmente os requisitos num nível de abstracção elevado, sendo tratados com maior detalhe *à posteriori*.

Os requisitos dos stakeholders são escritos segundo o ponto de vista dos stakeholders que pretendem o sistema, geralmente sem grande detalhe. É normalmente usada linguagem natural, diagramas informais ou anotações propícias para a resolução do problema. Os requisitos de sistema consistem em uma especificação detalhada, de forma abstracta segundo um modelo (diagramas de fluxos de dados, hierarquias de classes de objectos, etc.) complementada com notações em linguagem natural. Apesar de normalmente os requisitos de sistema serem mais detalhados, nada impede que o contrário se verifique ou que se definam num espaço intermédio entre os dois tipos [Sommerville & Sawyer, 1997].

Os requisitos constituem um conjunto multidimensional de necessidades. O objectivo não é ter no final um conjunto óptimo de requisitos, mas sim negociá-los entre todas as partes. Pretende-se um conjunto de requisitos que obtenha aprovação de todos, podendo ser necessário alterar alguns requisitos ao longo do tempo. Por outro lado, é essencial do ponto de vista de gestão, controlar os requisitos considerados críticos (requisitos que se não forem satisfeitos comprometem a existência de todo o sistema) [Gilb, 2003a].

Requisitos funcionais e não funcionais

De forma pratica, podemos dizer que os requisitos funcionais descrevem o que o sistema deve fazer, enquanto os requisitos não funcionais estabelecem constrangimentos sobre como os requisitos funcionais são implementados [Sommerville & Sawyer, 1997][Herzwurm et al., 1999]. Por exemplo, *implementar autenticação* é um requisito funcional e *em menos de 4 segundos* um requisito não funcional. No entanto um requisito funcional pode conter um requisito suplementar que diga que *a autenticação deve usar uma verificação da assinatura digital que tem de demorar menos de 4 segundos*, isto é, pode ser tanto um requisito funcional como não funcional. Geralmente os requisitos não funcionais com grande nível de abstracção são decompostos em requisitos funcionais de sistema [Sommerville & Sawyer, 1997].

Gilb (2003a) estabelece as seguintes categorias de requisitos:

- função: *o que* o sistema é capaz de fazer;
- qualidades: *quão bem* a função será desempenhada;
- custos: o que vai ser reportado de custos (dinheiro, pessoas, tempo, etc.) para criar e manter a função e suas qualidades;

- **constrangimentos:** quaisquer restrições à liberdade no processo de desenvolvimento.

Gilb (2003a) diferencia ainda entre dois tipos de requisitos:

- não quantificáveis (mas testáveis quanto à sua presença);
- quantificáveis (mensuráveis).

Na prática, é comum especificar a funcionalidade do sistema. No entanto, na especificação dos requisitos do sistema, são as categorias de requisitos *qualidade*, *custos* e *constrangimentos* que definem a condução do processo de desenvolvimento de software – na categoria *função* está implícito o que o sistema é capaz de fazer [Gilb, 2003a]. Por outro lado, é essencial que sejam definidos limites finitos quanto aos requisitos. De outra forma não poderiam ser enquadrados em níveis de qualidade.

No capítulo 3 será explorado com maior detalhe a distinção e aplicação das diferenças entre requisitos funcionais e não funcionais.

2.2.2 Processo de engenharia de requisitos

Os requisitos são a definição de *pontos de vista* dos stakeholders sobre o sistema. Estes pontos de vista entram no processo de desenvolvimento de software e originam um conjunto de soluções. As soluções encontradas em um nível de abstracção de uma etapa do processo de desenvolvimento servem de entrada no nível de abstracção seguinte. O processo de entrada e saída entre níveis de abstracção é aplicado no modelo QFD descrito no capítulo 3.

Gilb (2003a) sugere algumas recomendações de modo a promover a engenharia de requisitos:

- a) definir os requisitos claramente e sem ambiguidades;
- b) usar uma metodologia de especificação;
- c) fazer a ligação entre níveis de abstracção de forma clara;
- d) controlo rigoroso na qualidade de requisitos;
- e) não permitir especificações ambíguas ou não claras para serem usadas como entrada no nível de abstracção seguinte;
- f) não especificar detalhes demasiado cedo.

Problemas no processo de engenharia de requisitos

Sommerville & Sawyer (1997) apontam alguns problemas ao processo de engenharia de requisitos:

- normalmente ultrapassa o orçamento ou tempo planeado;
- as pessoas envolvidas não têm tempo / recursos suficientes para realizar as tarefas requeridas;
- existem queixas acerca do entendimento / conclusões do documento produzido;
- os stakeholders que desenvolvem o sistema queixam-se de trabalho inútil devido a erros nos requisitos;
- os stakeholders que utilizam o sistema falham em usar todas as capacidades do sistema;
- ocorre um grande volume de pedidos de alteração logo após a entrega do sistema aos stakeholders;

- demora muito tempo a implementar as alterações resultantes dos novos requisitos.

Priorização de requisitos

A definição de prioridades ajuda os stakeholders a definir as bases do sistema e a focalizar a atenção nas reuniões, especialmente se estiver associada uma análise de risco. A identificação explícita de requisitos prioritários ajuda igualmente os stakeholders responsáveis pelo desenvolvimento do sistema a decidir sobre a arquitectura e a resolver os conflitos que surjam.

Park et al. (2004) referem que um dos problemas mais comuns no desenvolvimento de software de forma colaborativa é priorizar os requisitos. É comum um projecto ter mais requisitos do que aqueles que pode de facto implementar face aos constrangimentos de tempo ou orçamento. Implementar requisitos debaixo destas condições e simultaneamente manter uma visão optimista de que é preciso satisfazer todos os stakeholders é absurdo. Nestas condições é inevitável que certos requisitos não serão implementados, ficando essa decisão para os stakeholders que desenvolvem o sistema, normalmente sobre limites de um prazo ou de um orçamento [Park et al., 2004].

Sommerville & Sawyer (1997) sugerem que as prioridades devem ser identificadas juntamente com a lista de requisitos, e que devem ser negociadas. Vulgarmente os stakeholders atribuem graus de importância diferentes para o mesmo requisito o que denota percepções e/ou necessidades diferentes. Madu (1994) refere ainda que essas diferenças devem ser entendidas.

Idealmente, a explicitação das prioridades deveria ser feita durante a extracção de requisitos. Pode-se tornar no entanto difícil quantificar prioridades ou mesmo

implicações sem uma visão global dos requisitos, sendo neste caso razoável fazer uma avaliação depois da percepção do que está envolvido.

Pode acontecer que os stakeholders classifiquem todos os seus requisitos como *essenciais*. O principal problema, será justificar perante os stakeholders que pretendem o sistema (especialmente os de topo) que os seus requisitos são desnecessários, pelo que devem ser confrontados por exemplo, com os custos do desenvolvimento desses requisitos.

Avaliação dos riscos

Para cada requisito, ou conjunto de requisitos relacionados, deverá ser coleccionado um conjunto de possíveis problemas que poderão surgir na implementação, a probabilidade de acontecerem e as suas implicações. A avaliação explícita do risco é uma forma de identificar problemas de implementação que poderão ser minimizados se detectados nesta fase, pela modificação dos requisitos que lhe dão origem. Geralmente os requisitos que apresentam maior risco são aqueles dos quais se tem menos informação, pelo que necessitam de ser esclarecidos pelos stakeholders [Sommerville & Sawyer, 1997].

A avaliação dos riscos não pode ser feita de forma quantitativa, ou seja, não podem ser atribuídos valores de risco. Devem ser utilizadas técnicas do tipo lógica difusa na sua categorização tais como *alto*, *médio* e *baixo*. A dificuldade na aplicação da análise de risco prende-se com a identificação dos tipos de risco inerentes à própria organização. Devem ser por isso envolvidas pessoas com experiência em análise de risco na avaliação de cada requisito. Um problema que se coloca por vezes é que inicialmente não existem pessoas com experiência neste tipo de análise, pelo que vai sendo ultrapassado à medida que os stakeholders vão ganhando experiência [Sommerville & Sawyer, 1997].

Listas para análise de requisitos

O objectivo da análise de requisitos é descobrir problemas e não gerir a qualidade das actividades. Assim sendo, a lista não deverá conter questões relacionadas por exemplo com a conformidade com as normas da organização. O processo é simples: o analista assinala os itens da lista enquanto consulta o documento de requisitos. Quando é notado um potencial problema, este é referido juntamente com o requisito.

As listas devem ser elaboradas de acordo com a maturidade da organização. No nível inicial, uma lista com dez itens é suficiente, mas revela-se pobre para uma organização mais madura. A criação das listas torna o processo de análise mais moroso que um processo de análise informal, exigindo ainda uma gestão constante dos itens considerados de forma a que se mantenham aptos a uma contra-análise, mas analisa a conformidade dos requisitos e permite descobrir problemas.

As listas devem ser revistas no final de cada projecto. Quando são descobertos problemas novos, devem ser analisadas as listas de forma a concluir se novas listas permitiriam detectar esses problemas. Deve-se ter em atenção que essas listas assentam na política da organização pelo que requerem autorização para serem alteradas.

A análise de requisitos baseada em listas deve ser usada com cuidado dado que a excessiva focalização neste método tende a restringir os stakeholders aos pontos descritos nas mesmas, sem um olhar crítico de outros problemas não contemplados nos itens (especialmente se forem obrigados a assinalar os itens no processo de análise de requisitos). Deve ser entendido pelos stakeholders que as listas servem como um guia de forma a que os itens relevantes para a organização não sejam esquecidos, mas não devem servir de restrição para a procura de eventuais problemas.

No capítulo 5 procura-se dar suporte a listas para análise de requisitos.

Princípios fundamentais da engenharia de requisitos

Gilb (2003a) descreve os seguintes princípios fundamentais da engenharia de requisitos:

- todos os requisitos são testáveis quanto à sua presença no contexto da implementação;
- alguns requisitos são binários, outros escalares;
- todos os requisitos reflectem valores e prioridades subjectivas de alguém;
- para qualquer conjunto de requisitos existe outro conjunto de requisitos que provavelmente satisfaz a necessidade, valor e prioridade dos intervenientes, melhor ou de forma igual;
- todos os requisitos estão em conflito natural com todos os outros que partilhem recursos comuns;
- os requisitos não são estáticos, dado que o contexto envolvente afecta necessidades, valores e prioridades;
- a engenharia de requisitos é um processo sistemático de determinação do conjunto relevante de valores considerados pelos stakeholders, até que seja atingido o nível de satisfação pretendido.

2.3. Negociação de Requisitos

A negociação de requisitos é uma das fases iniciais de qualquer ciclo de desenvolvimento de software, mas o seu impacto reflecte-se a nível global do sistema.

2.3.1 Análise de Requisitos e Negociação

Depois de se obter um conjunto de requisitos, deve ser efectuada uma análise cuidada de forma a procurar conflitos, sobreposições, omissões e inconsistências. Após esta etapa, os stakeholders negociam a aceitação de um conjunto de requisitos. Os conflitos devem ser resolvidos e os requisitos ordenados pela sua importância, como referido anteriormente.

O objectivo da análise e negociação de requisitos é o de estabelecer uma aceitação sobre um conjunto de requisitos completos e consistentes onde devem ser resolvidas quaisquer ambiguidades. Idealmente durante este processo são descobertos requisitos que faltam, conflitos entre requisitos (o que exige que se tomem decisões atendendo aos benefícios face aos custos), ambiguidades, sobreposições, requisitos irrealistas, etc.

A recolha, análise e negociação de requisitos estão portanto intimamente ligados e fazem parte de um processo em espiral. Este processo deve ser seguido até à satisfação de todas as partes envolvidas, sendo um processo influenciado não só pela lógica e argumentos técnicos mas também pela política da organização e pela personalidade dos stakeholders [Sommerville & Sawyer, 1997].

2.3.2 Planeamento dos conflitos e resolução

As organizações não costumam gastar muito tempo a resolver os conflitos entre requisitos, provavelmente porque existe a convicção de que os conflitos são um mal necessário, não parecendo razoável resolver esse tipo de falha. Dado que os conflitos nascem dos diversos pontos de vista (necessidades, prioridades) entre diferentes stakeholders, deve existir um espaço onde se possa negociar – não é possível ter um sistema que agrade *cem por cento* a todos, e se não houver oportunidade de se chegar a

um acordo, vai ser formado um conjunto de pessoas inimigas do sistema para as quais o mesmo foi criado.

As reuniões são o meio mais rápido para resolver conflitos, e onde devem estar presentes os stakeholders que detectam as falhas e os stakeholders que as resolvem – apesar de os sistemas electrónicos serem eficazes na troca de informação, não substituem o contacto directo onde são tomados compromissos [Costa et al., 1998].

Invariavelmente o conflito, a sobreposição e a omissão de requisitos estarão presentes na análise dos mesmos pelo que se torna necessário realizar um planeamento prévio das reuniões. Se realizado, focaliza-se a atenção na resolução de conflitos em vez de ser perdido tempo na obtenção do conjunto de requisitos a debater. Por outro lado, com o planeamento garante-se que as diferentes opiniões dos intervenientes são discutidas fazendo aumentar a confiança dos stakeholders [Costa et al., 1998].

O modelo QFD descrito no capítulo 3 é propício ao planeamento prévio das reuniões de negociação, nomeadamente à construção do conjunto de alternativas.

2.3.3 Reuniões de Negociação

Sommerville & Sawyer (1997) sugerem que a reunião de negociação seja conduzida da seguinte forma (3 estados):

- 1) informação: onde se faz o enquadramento dos problemas afectos ao requisito;
- 2) discussão: onde se debate a forma de como os problemas podem ser resolvidos, sugerindo-se que exista um tempo estabelecido para cada item para que se garanta que todos eles são discutidos, sem esquecer a participação de todos os stakeholders;
- 3) resolução: onde se chega a um consenso.

Estes três estados estão presentes em qualquer processo de negociação. Estes estados são desenvolvidos na forma de *pré-negociação*, *negociação* e *pós-negociação* no capítulo 5.

Quando se tem pouco tempo para dispensar em reuniões é conveniente por um lado partir um reunião longa em várias mais curtas e por outro lado identificar os requisitos de maior importância (maior prioridade, mais críticos) para serem analisados em primeiro lugar. Apesar de se gastar mais tempo a preparar a reunião, garantimos que pelo menos os assuntos mais importantes são tratados [Sommerville & Sawyer, 1997].

Uma das questões importantes na negociação é determinar o conjunto de decisões alternativas. Este problema é geralmente resolvido antes da negociação começar, ou seja, os stakeholders constroem uma lista (conjunto) de todas as alternativas. É possível, no entanto, que o problema do que é ou não possível, ou em geral, o que é ou não uma alternativa, seja isso mesmo uma questão de negociação. Por outro lado, os stakeholders têm de determinar um espaço aceitável e comum para que exista comunicação. O espaço comum é normalmente considerado o espaço de alternativas, de acordo com os requisitos e desejos pessoais. O problema é escolher uma alternativa como compromisso [Kersten, 1987].

No modelo QFD descrito no capítulo 3, a lista de todas as alternativas é elaborada a partir dos requisitos dos stakeholders que vão utilizar o sistema, antes das reuniões de negociação. As alternativas que suscitam conflito e que não foram resolvidas antes da reunião são debatidas pessoalmente entre os stakeholders na reunião de negociação.

2.4. Conclusões

O facto de os stakeholders não terem um ponto de vista único resulta na necessidade de analisar e negociar requisitos de software que estão em conflito (identificação,

representação, análise, gestão e integração dos diferentes pontos de vista). A negociação torna-se uma parte essencial da especificação do sistema: os stakeholders que utilizam o sistema negociam entre eles e com os stakeholders que o desenvolvem, sendo necessário fazer cedências para a resolução dos conflitos.

Os resultados da negociação podem ser avaliados de duas formas: o que são resultados justos e a forma como o processo é conduzido [Lewicki & Litterer, 1985]. Dado que em situações de negociação não existem padrões claros que definam os resultados da uma negociação justa, o processo ganha uma maior importância, ajudando a transmitir uma imagem de igualdade, justiça e reciprocidade nos propósitos e concessões. Contornar este processo poderá ser contraprodutivo, conduzindo à desilusão e irritação mesmo quando os resultados são favoráveis. Um processo justo implica um reconhecimento da posição, estatuto e legitimidade – uma consideração pouco enfatizada, onde o foco se centra na resolução do problema evitando o processo de dar e receber [Chaudhury et al., 1991].

Não é possível melhorar a qualidade da engenharia de software enquanto não se melhorar a habilidade de articulação de requisitos sem ambiguidades e de forma clara, e enquanto não for entendida a diferença entre objectivos e meios de os atingir [Gilb, 2003a].

O modelo QFD descrito no capítulo 3 estabelece a diferença entre objectivos e os meios de os atingir em todas as etapas do processo de desenvolvimento de software. A priorização dos requisitos é um ponto chave do modelo e associado à solução apresentada no capítulo 5, permite uma estratégia de negociação intermédia entre integrativa e distributiva.

Capítulo 3

Modelos

3.1. Contexto

Apesar das metodologias tradicionais de desenvolvimento de sistemas reconhecerem a importância dos stakeholders, elas não fornecem métodos formais que transformem os requisitos desses stakeholders em especificações de desenho do sistema. Por outro lado, as metodologias não são rigorosas a manter a atenção nos requisitos definidos pelos stakeholders que vão utilizar o sistema, nas fases posteriores de desenvolvimento [Stylianou et al., 1997].

Uma técnica utilizada em organizações com programas de *Gestão de Qualidade Total* (Total Quality Management, TQM) [Zultner, 1993] [Stylianou et al., 1997] para focar a atenção nos requisitos dos stakeholders que vão utilizar o sistema por todo o processo de desenho, produção e entrega dos produtos denomina-se *Desdobramento da Função Qualidade* (Quality Function Deployment, QFD) [Zultner, 1993], tendo sido aplicada mais tarde ao processo de desenvolvimento de software (Software Quality Function Deployment, SQFD) [Haag et al., 1996].

O principal objectivo do QFD é o envolvimento dos stakeholders que vão utilizar o sistema durante todo o processo de desenvolvimento, de modo a ser obtido um sistema de elevada qualidade em que os stakeholders se sintam satisfeitos. Uma das principais razões para os stakeholders se sentirem insatisfeitos é que os seus objectivos de

qualidade não são cumpridos. É por isso imprescindível captar esses requisitos, numa forma que se expresse em especificações detalhadas de desenho, para que desde o início se trabalhe em função desses requisitos, numa metodologia mais orientada para os stakeholders [Stylianou et al., 1997].

O QFD pode ser definido como *um desdobramento passo-a-passo de uma função ou operação que implica qualidade, em detalhes através da sistematização de alvos e significados* [Akao, 1997].

Em meados de 1960, no Japão, o Dr. Akao apresentou o QFD, numa época em que as indústrias passaram da imitação de produtos para o desenvolvimento baseado na *originalidade*. Actualmente, no domínio do software, a qualidade exigida é cada vez maior pelo que através do QFD se consegue [Akao, 1997]:

- identificar essa qualidade;
- um método para a construção de sistemas que assegurem essa qualidade.

As metodologias tradicionais de desenvolvimento de software normalmente começam por uma especificação não estruturada do problema e produzem uma especificação detalhada do desenho, quase sempre por um processo sucessivo de refinamento. O principal foco dessas metodologias são os dados e o processo de modelação, em que têm em comum [Stylianou et al., 1997]:

- começar com um problema não estruturado; modelar as características do problema recorrendo a construções que são dependentes da metodologia; refinar o modelo até se chegar a um nível de detalhe que seja possível implementar;
- fazer decisões de desenho e implementação do sistema usando uma tecnologia;
- desenvolver e implementar o sistema.

No entanto, um nível de detalhe cuidado e o envolvimento dos stakeholders pode não ser suficiente para garantir um nível elevado de qualidade [Stylianou et al., 1997]. Os métodos tradicionais não garantem:

- que sejam incluídos todos os stakeholders nem os seus requisitos de qualidade;
- que os requisitos de qualidade sejam sistematicamente considerados pelas várias etapas do processo de desenvolvimento;
- a priorização dos requisitos dos stakeholders.

O QFD é uma ferramenta de TQM que na sua forma mais básica em cada etapa do processo de desenvolvimento transforma os requisitos dos stakeholders em requisitos técnicos. No entanto, não fornece modelos de construção de dados, processos ou objectos. Os modelos de construção são extremamente úteis na passagem de modelos de sistemas para código-fonte. Stylianou et al. (1997) sugere portanto que é na *integração* do QFD com metodologias de modelação de sistemas que se torna possível conseguir sistemas direccionados para a qualidade. Não se pretende substituir os processos tradicionais pelo modelo QFD, mas sim integrar os processos descritos a um nível de abstracção superior (TQM).

3.2. Desdobramento da Função Qualidade

O Desdobramento da Função Qualidade (QFD) distingue entre requisitos e solução de implementação, podendo os seus objectivos ser resumidos à [Stylianou et al., 1997]:

- identificação dos stakeholders;
- identificação do *que* desejam;
- identificação de *como* satisfazer esses desejos.

O princípio de distinguir requisitos e soluções pode ser estendido a todo o processo de desenvolvimento, dado que os resultados de um processo correspondem aos requisitos do seguinte. Significa não só separar *o que* é para ser realizado de *como* é que será implementado, mas também fazendo distinção entre *o que* e as razões do *porquê* de algo ter de ser realizado. Apenas as respostas a este *porquê* descrevem os benefícios na forma de vantagens competitivas quando os clientes usarem o produto, possibilitando soluções inovadoras. É ainda essencial que sejam definidos limites finitos quanto aos requisitos de modo a serem enquadrados em níveis de qualidade, conforme referenciado no capítulo 2.

A evolução do QFD levou ao aparecimento de várias versões da metodologia [Akao, 1997]. O QFD-estendido é uma metodologia QFD que serve de ilustração às aproximações mais comuns das metodologias que aparecem na literatura [Mazur, 1993][Hierholzer et al., 1998][Herzwurm et al., 1999] [Knogstie, 1999].

Em [Peixoto & Carpinetti, 1998] vem descrito o QFD-estendido, caracterizado pela existência de quatro fases (planeamento de produto, planeamento de componentes, planeamento de processo e planeamento de produção) (figura 3.1). Na primeira fase existe a uma matriz denominada Casa da Qualidade (House of Quality, HoQ) que transforma os requisitos dos stakeholders que pretendem usar o sistema em elementos de qualidade. Os elementos de qualidade mais importantes desta matriz são relacionados numa segunda matriz com as características dos componentes do possível produto. Um processo análogo é seguido para as matrizes seguintes, tendo em atenção os *falsos* requisitos definidos no capítulo 2.

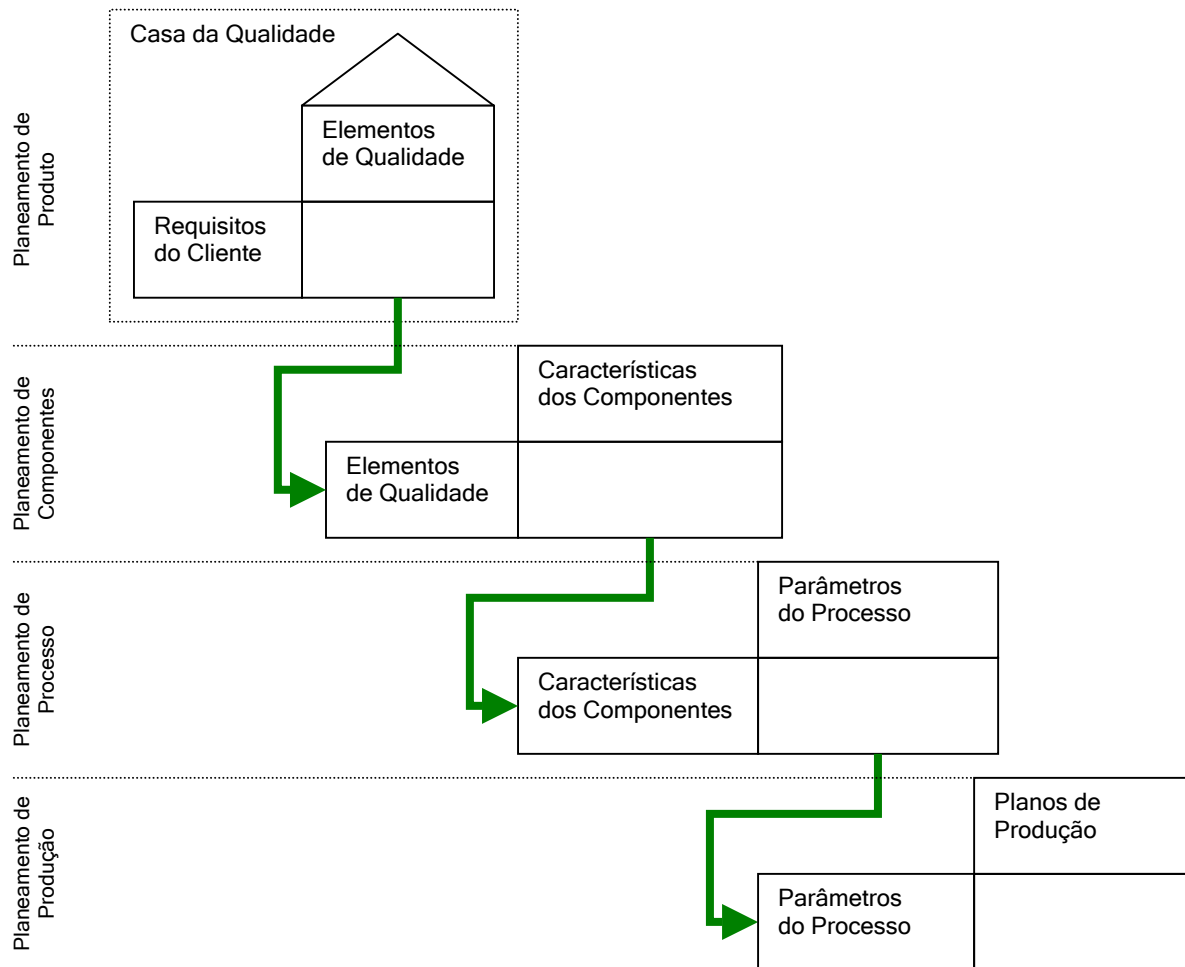


Figura 3.1. QFD-estendido em quatro fases
 (adapt. de [Peixoto & Carpinetti, 1998])

No QFD-estendido, os requisitos dos stakeholders entram na fase de planejamento do produto, são processadas durante as quatro fases e saem na fase de planejamento de produção como informações para o desenvolvimento do sistema – as matrizes têm sempre uma entrada que é a saída da matriz anterior. Este processo explicita a tomada de decisão, dado que as decisões tomadas na matriz anterior [Peixoto & Carpinetti, 1998]:

- determinam as decisões na matriz corrente;
- direccionam a concretização das decisões da matriz precedente.

O QFD promove o desdobramento dos requisitos / soluções em múltiplas matrizes que derivam umas das outras num processo semelhante ao refinamento das metodologias tradicionais, conforme a figura 1.3 do capítulo 1. A distinção entre *porquê* e *o quê* mostra-se nos requisitos dos stakeholders que pretendem o sistema e do produto, enquanto a independência da implementação descreve a separação entre *o quê* e *como* [Herzwurm et al., 2002].

O processo de QFD é sustentado por uma equipa QFD formada por stakeholders de todos os departamentos (desenvolvimento de software, gestão, etc.) e que se reúne com os stakeholders que pretendem o sistema. As necessidades dos stakeholders são recolhidas e classificadas numa tabela denominada Voz do Cliente (Voice of Customer, VoC). Esses requisitos são estruturados com recurso a Diagramas de Afinidade [Hierholzer et al., 1998], avaliados segundo o *Analytical Hierarchy Process* (AHP) e obtido o nível de satisfação de cada requisito no caso de implementação, conseguindo assim a Tabela de Requisitos do Cliente, num processo assistido pelos stakeholders que pretendem o sistema (figura 3.2).

Um aspecto importante no QFD é que a qualidade é definida pelos stakeholders que pretendem o sistema usando as suas próprias palavras. Tradicionalmente *qualidade de software* e *sistemas* em geral são definidos em termos técnicos que podem significar pouco ou nada para os stakeholders que pretendem o sistema [Stylianou et al., 1997]. A sua relativa facilidade de uso e compreensão, a habilidade de transmitir informação de forma sistemática entre etapas e o fornecimento de uma base para identificar conflitos entre requisitos são pontos a favor do QFD [Stylianou et al., 1997].

É necessário identificar as características de produto que servirão de entrada para a Casa da Qualidade. O processo é análogo à obtenção da Voz do Cliente, só que agora recorrendo à Voz do Engenheiro (Voice of Engineer, VoE) (figura 3.2). Os stakeholders que pretendem o sistema não estão presentes nestas reuniões.

As correlações nas matrizes são expressas na forma de números { 0, 1, 3, 9 } que representam o grau de correlação entre a satisfação de um requisito por um elemento de qualidade. No caso da matriz Casa de Qualidade, a partir das correlações definidas são obtidos pesos para cada requisito / solução, permitindo ordenar pela sua importância (priorizar) tanto os requisitos como as soluções da matriz.

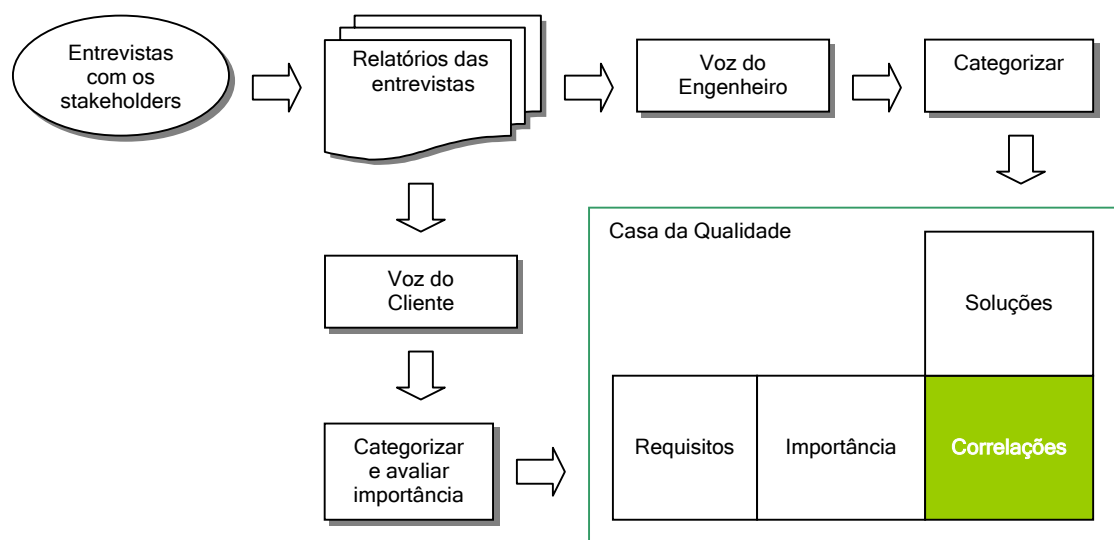


Figura 3.2 – Esquema simplificado da obtenção de requisitos e soluções da Casa da Qualidade

Stylianou et al. (1997) resumem as etapas principais do QFD em:

- identificação dos stakeholders;

-
- estabelecimento dos requisitos (atributos) dos stakeholders para o produto ou serviço e relativa importância;
 - identificação das características do produto ou serviço;
 - compreensão da correlação entre as características do produto / serviço e os requisitos dos stakeholders;
 - comparação das características de produto com a concorrência;
 - identificação dos pontos atractivos do produto / serviço relativamente à concorrência;
 - desenvolvimento de valores alvo para as características do produto / serviço final baseados na importância dos stakeholders, pontos atractivos e avaliação competitiva.

A figura 3.3 apresenta as principais áreas que podem ser encontradas na Casa da Qualidade, seguindo-se uma breve descrição sobre cada uma.

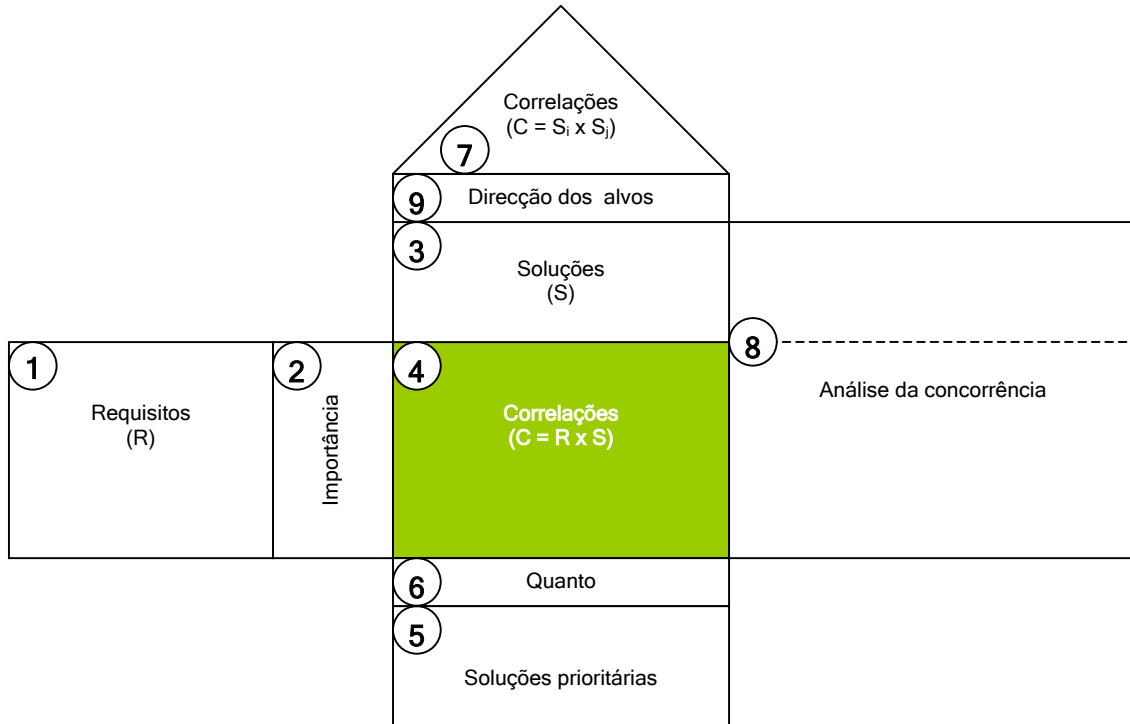


Figura 3.3 – Esquema das principais áreas da Casa da Qualidade

Área 1 Requisitos

Descreve as necessidades dos stakeholders que pretendem o sistema. São usados termos em linguagem natural.

Área 2 Importância

Os requisitos são avaliados segundo o *Analytical Hierarchy Process* (AHP).

Área 3 Soluções

Descreve as características do sistema segundo a perspectiva dos stakeholders responsáveis pela sua implementação, baseados nas necessidades dos stakeholders que pretendem o sistema.

Área 4 Correlações

Representam a força da relação entre a área 1 e a área 3, segundo a perspectiva dos stakeholders.

Área 5 Soluções prioritárias

A figura 3.4 representa as áreas 1 a 5 da figura 3.3. As soluções prioritárias são calculadas a partir da importância *i* e das correlações *c* obtidas.

Os pesos das soluções *s_n* da figura 3.4 são obtidos através de:

$$s_n = \sum (i_j * c_{jn}), n = 1, 2, 3, j = 1, 2, 3$$

	importância do requisito	solução 1	solução 2	solução 3
requisito 1	i1	c11	c12	c13
requisito 2	i2	c21	c22	c23
requisito 3	i3	c31	c32	c33
		s1	s2	s3

Figura 3.4. Cálculo de soluções prioritárias

Área 6 Quanto

Definem-se os valores alvo a atingir (valor e unidade de medida).

Área 7 Correlações

Esta área identifica o impacto das soluções entre si, ajudando na resolução de conflitos entre soluções.

As relações possíveis entre soluções são { POSITIVO_FORTE | POSITIVO | NEGATIVO | NEGATIVO_FORTE }.

Área 8 Análise da concorrência

É feita uma avaliação dos requisitos da área 1 relativamente às soluções da concorrência, com o fim de obter um sistema que:

- exceda os produtos da concorrência;
- ofereça soluções inovadoras.

Área 9 Direcção dos alvos

Esta área revela se os valores da área 6 devem aumentar, diminuir ou atingir um valor específico.

Haag et al. (1996) refere que o uso mais comum do QFD na indústria de sistemas de informação parece ser na análise de requisitos de software.

3.3. Desdobramento da Função Qualidade no Software

O QFD original, sendo usado inicialmente na indústria, relaciona as características do produto com características de qualidade mensuráveis. O QFD aplicado ao

desenvolvimento de software implica um acréscimo ao modelo QFD. O software é identificado não por características físicas mas pela funcionalidade, significando que os elementos de qualidade do QFD clássico são substituídos por funções do produto no SQFD. Por outro lado, distingue-se também entre características funcionais (funções do produto) e não funcionais (elementos de qualidade) [Sommerville & Sawyer, 1997] [Gilb, 2003a] conforme descrito na tabela 3.1.

Tabela 3.1. Definições importantes no SQFD
(adapt. de [Herzwurm et al., 1999])

	Necessidades dos stakeholders que pretendem o sistema = exigência	Característica do produto = solução	
Definição	Necessidade resultante do uso do produto	Características ou capacidades do produto, independentes da implementação	
Expressão	Requisito	Funções do produto	Elemento de qualidade
Forma	Curto, conciso, em linguagem natural	Características funcionais do produto, geralmente não mensuráveis	Características não funcionais, possivelmente mensuráveis durante o desenvolvimento e depois de concluído

Relaciona-se em primeiro lugar a Voz do Cliente com as características funcionais obtidas a partir da Voz do Engenheiro. Incluir explicitamente a Voz do Engenheiro na forma de função de produto é importante inclusivamente para serem identificados factores que excedam as expectativas dos clientes, i.e, características de software que eles próprios não tenham mencionado. Como resultado, tem-se a substituição dos elementos de qualidade da primeira matriz (Casa da Qualidade *clássica*) por funções de produto, o que origina uma nova matriz denominada Casa da Qualidade para Software (software-HoQ).

Em segundo lugar, relacionamos os requisitos de cliente com os elementos de qualidade na Casa da Qualidade clássica (figura 3.5).

Ao serem relacionadas as tabelas *Requisitos do Cliente e Funções de Produto* na Casa da Qualidade para Software, obtemos como saída uma tabela das *Funções de Produto Mais Importantes* (FPMI). Na Casa da Qualidade clássica, da relação de tabelas *Requisitos do Cliente* com *Elementos de Qualidade*, obtemos como saída uma tabela de *Elementos de Qualidade Mais Importantes* (EQMI). Relaciona-se numa terceira matriz as (FPMI x EQMI) como base para um desdobramento mais pormenorizado, representando uma análise de desenho da aplicação (planeamento da funcionalidade).

A figura 3.6 apresenta um exemplo do desdobramento de um requisito no QFD aplicado ao software.

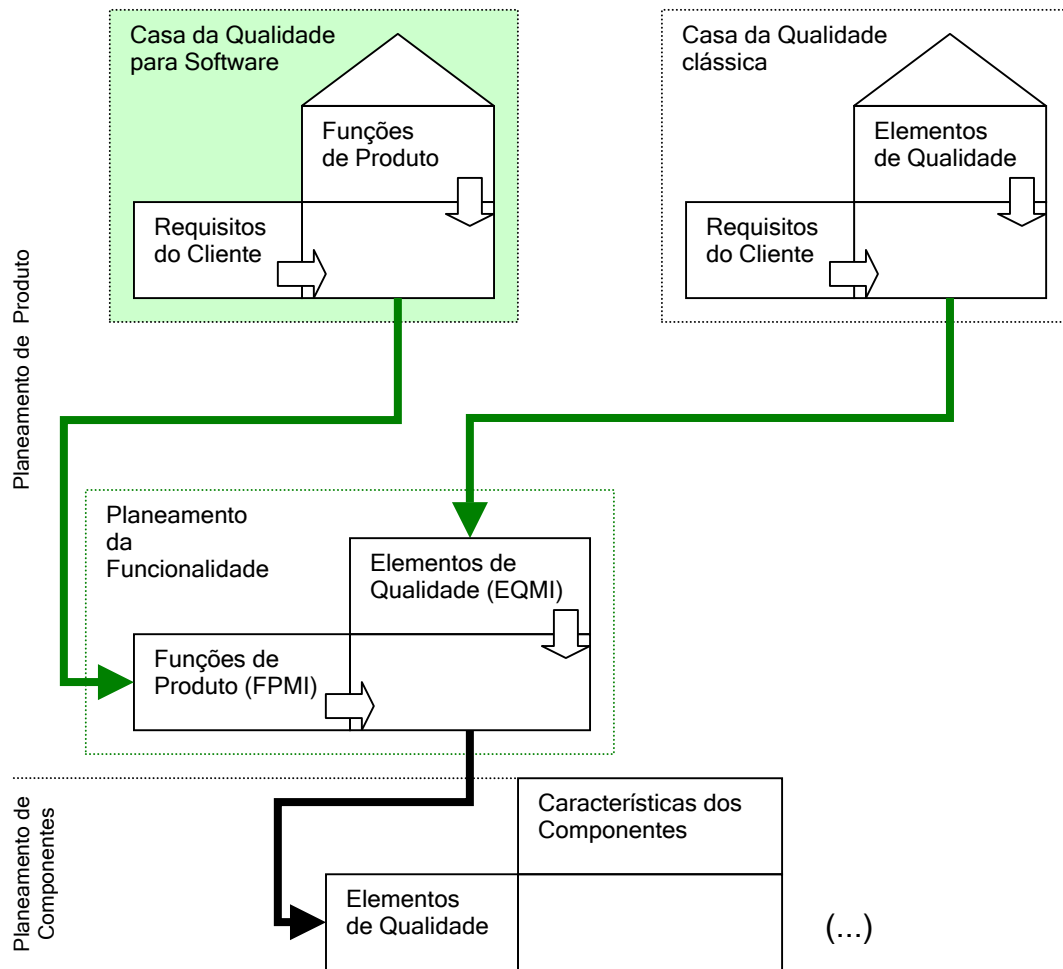


Figura 3.5. QFD aplicado ao Software

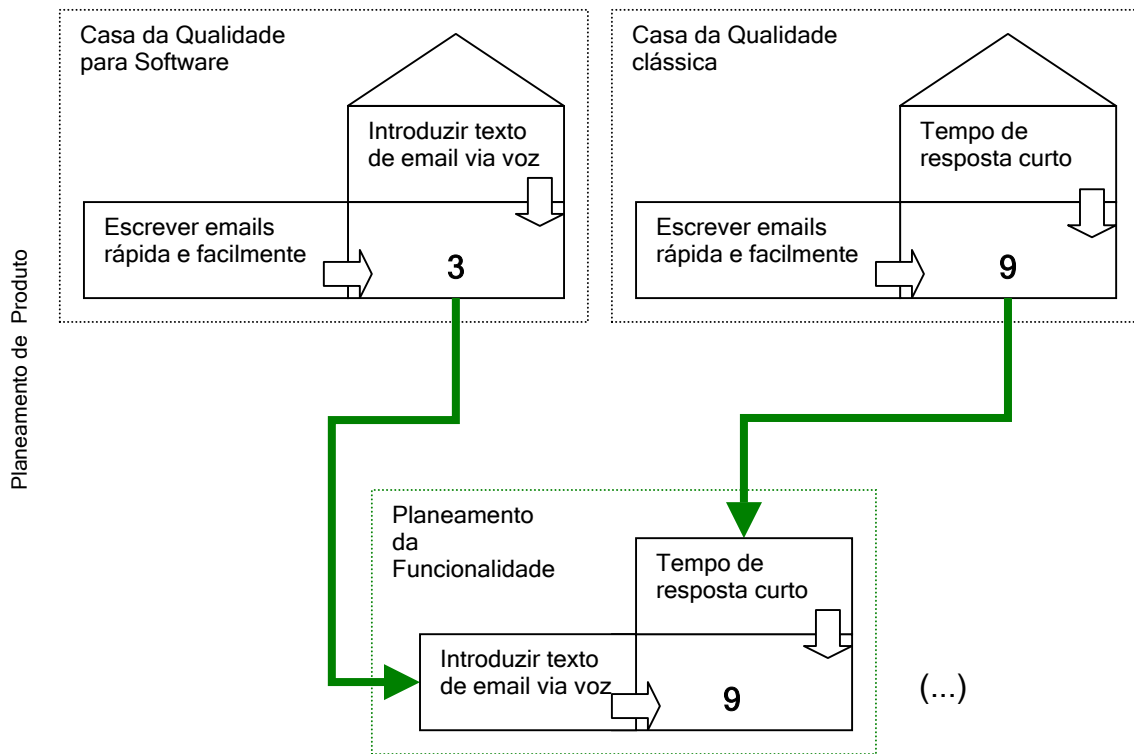


Figura 3.6. Exemplo de um requisito do Cliente no SQFD

3.4. IBIS

O sistema de argumentação mais popular, *Issue Based Information System* (IBIS) [Conklin & Begeman, 1988] utiliza *questões* (issues), *posições* (positions) e *argumentos* (arguments), para mediar a conversação entre stakeholders. Basicamente, no modelo IBIS existe uma *questão* que serve de base à discussão onde as *posições* dos vários stakeholders reflectem as opiniões a favor ou contra a *questão* e os *argumentos* suportam as posições. O modelo IBIS pode ser usado para representar um processo de discussão através do QFD [Jacobs & Kethers, 1994].

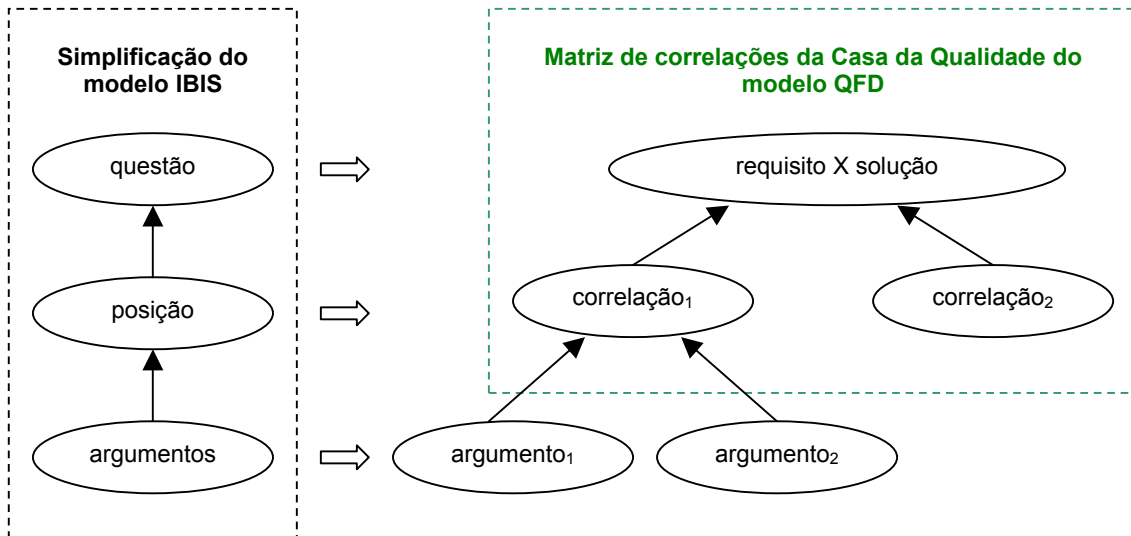


Figura 3.7. Esquema simplificado do modelo IBIS e correspondência com a Casa da Qualidade do modelo QFD (adapt. de [Reich, 1996])

Reich (1996) faz uma analogia entre o modelo IBIS e as matrizes QFD:

- a *questão* corresponde ao par { Requisito, Solução } de uma matriz;
- a *posição* é representada por um valor de correlação de { 0, 1, 3, 9 }.

A solução proposta no capítulo 5 utiliza uma analogia semelhante a Reich (1996) em que:

- a *questão* pode ser apresentada na forma “Qual o valor da correlação que relaciona o requisito com a solução?”;
- a *posição* implícita de “a favor” ou “contra” a partir dos valores explícitos de correlação dos stakeholders;
- os *argumentos* suportam as posições.

As posições não aparecem com valores explícitos na solução proposta devido às estratégias de negociação (são apresentadas apenas as posições implícitas). A figura 3.7 mostra um esquema simplificado do modelo IBIS e sua correspondência com a Casa da Qualidade do modelo QFD.

O suporte à argumentação visa aumentar a racionalidade do processo de negociação, aumentando inclusive a memória organizacional pelo registo dos argumentos que levam a uma determinada posição relativamente a uma questão que dá origem ao processo. No entanto, o suporte à argumentação restringe-se a uma iteração do processo de negociação que começa quando existe conflito e termina quando o conflito é resolvido. Para um novo processo de negociação se for pretendido justificar as posições assumidas torna-se necessário inserir novamente os argumentos.

Na solução proposta os stakeholders não podem escolher livremente os argumentos, sendo a escolha restringida a um vocabulário de termos. Com este procedimento, evita-se a verbalização e correspondente dispersão de argumentos [Hwang & Lin, 1987].

3.5. Ontologia dos argumentos

No capítulo 2 é referido que para cada requisito deverá ser coleccionado um conjunto de possíveis problemas – avaliação explícita do risco. Esta análise pode ser feita confrontando cada requisito com uma lista baseada na experiência, permitindo inclusivamente a reutilização de conhecimento de outros projectos. Esta lista pode ser baseada num *conjunto de palavras reservadas que fazem sentido num determinado contexto (ontologia)* definida por uma organização e que fundamenta a *posição* assumida face a uma *questão*.

Nour (2003) faz referência a duas definições de ontologia:

- especificação explícita de uma conceptualização;

- vocabulário de termos definidos numa forma que pode ser lida por pessoas e máquinas.

Entende-se conceptualização como uma *abstracção ou visão simplificada de uma área de conhecimento*. Esta abstracção permite aos stakeholders:

- partilhar conhecimento;
- compreender mutuamente uma área de conhecimento;
- atingir um consenso sobre uma área de conhecimento.

Ontologia pode ser entendida como [Nour, 2003]:

- um método formal de descrição de conceitos e suas relações dentro de um certo domínio;
- uma especificação que determina o numero de termos, relações e afirmações que podem estar contidas num determinado domínio.

O conceito pode ser estendido à construção de *bases de conhecimento* que descrevam uma determinada área de conhecimento. Uma base de conhecimento é uma colecção de conhecimento adquirido a partir da experiência e guardado num repositório. Estas bases de conhecimento permitem uma comunicação entre stakeholders em que não é necessário recorrer a conceitos que se encontrem fora dos limites definidos pela ontologia [Nour, 2003].

Neste trabalho, é sugerido que os argumentos devem fazer parte de uma terminologia que seja partilhada por todos os stakeholders de modo a que [Lin et al., 1996]:

- a comunicação seja realizada no mesmo contexto e permita elevados níveis de colaboração;
- a precisão e não ambiguidade evite possíveis conflitos e interpretações diferentes.

Uma dificuldade na definição da ontologia prende-se com o facto de ser inerente a cada organização, devendo por isso ser definida por pessoas da organização.

3.6. Consenso e Voto Maioritário

O Voto Maioritário consiste num *processo de decisão, baseado em votação onde a vontade do maior grupo prevalece*.

Ao invés, o Consenso é *um processo de decisão que trabalha de forma criativa na inclusão das opiniões de todos os stakeholders na elaboração da decisão*. Em vez de simplesmente se votar cada *questão* e obter o voto maioritário do grupo, é promovido o encontro de soluções com as quais *todos* possam viver. Procura-se assegurar que são consideradas as opiniões, ideias e reservas de todos os stakeholders e onde as decisões são efectuadas num diálogo entre iguais: cada stakeholder tem o direito a sugerir os valores pretendidos e prevenir alterações que julguem inaceitáveis (o direito de *vetar* uma decisão significa que as minorias não podem ser ignoradas). As *questões* são discutidas até que todos cheguem a um acordo [Seedsforchange, 2004].

Note-se que o consenso é mais do que um compromisso. Trata-se de um processo do qual podem resultar soluções criativas que excedam as soluções propostas inicialmente.

O voto maioritário induz um tipo de interacção diferente do usado no consenso: quando os participantes compreendem que podem construir coligações quando não chegam a

acordo, o foco deixa de estar concentrado no encontro de soluções que agradem a todos, implicando inclusivamente soluções de valor inferior.

No consenso, o grupo de stakeholders é que toma a decisão, necessitando cada stakeholder por um lado, de tentar persuadir a outra parte acerca das suas próprias necessidades e interesses, e por outro, ouvir atentamente para a determinação de como a solução pode preencher as necessidades dos outros [Maine, 2004].

A tabela 3.2 resume alguns pontos comparativos entre consenso e voto maioritário.

Tabela 3.2 Pontos comparativos entre consenso e voto maioritário

Consenso	Voto maioritário
Abordagem win-win	Abordagem win-lose
Promove cooperação	Promove competição
Todos os stakeholders tem o mesmo poder de influenciar a decisão	O poder de decisão corresponde a hierarquia, popularidade, etc.
Os stakeholders concordam ou não concordam	Os stakeholders votam contra, a favor ou abstêm-se
Todas as ideias são ouvidas e exploradas	A maioria decide quais as ideias a serem consideradas

O stakeholder responsável pelo projecto facilita o processo procurando garantir que todas as partes são consideradas	O stakeholder responsável pelo projecto controla o processo e determina quem deve ser considerado
Um stakeholder pode inviabilizar a decisão	São necessários 51% dos stakeholders para inviabilizar a decisão
Não existem minorias; exige aceitação completa	A maioria ganha e a minoria perde
Todos os stakeholders aceitam inteira responsabilidade pelas decisões do grupo	Os stakeholders podem descartar responsabilidades alegando que não votaram a favor
O processo nem sempre conduz a uma decisão	O processo conduz sempre a uma decisão
O processo de decisão pode ser demorado	O processo de decisão pode ser rápido
O processo de aceitação da implementação pode ser rápido dada a aceitação completa da decisão	O processo de aceitação pode ser demorado dado que 49% dos stakeholders podem estar contra

Uma análise à tabela 3.2 sugere que para uma negociação ser bem sucedida, a estratégia a seguir deve ser baseada em consenso. No entanto, o voto maioritário pode ajudar a medir a aproximação a um consenso, significando que pode ser considerado um mecanismo de ajuda à negociação. Tem-se que a utilização deste mecanismo fornece suporte à resolução do conflito por consenso na avaliação de requisitos de software fornecendo simultaneamente informação quanto à preferência dos stakeholders.

3.7. Conclusões

No âmbito de uma Gestão de Qualidade Total, o Desdobramento da Função Qualidade proporciona níveis de satisfação dos stakeholders muito superiores aos alcançados pelas metodologias tradicionais [Stylianou et al., 1997].

No domínio do software, a qualidade exigida é cada vez maior pelo que através do QFD se consegue [Akao, 1997]:

- identificar essa qualidade;
- um método para a construção de sistemas que assegurem essa qualidade.

No QFD são usadas matrizes em que se relacionam requisitos / soluções sem mostrar explicitamente os objectivos dos stakeholders. Importa por isso expor as razões que sustentam uma *posição* de um stakeholder face a um valor inserido por outro stakeholder no processo de negociação. Uma solução será proporcionar suporte à argumentação tipo IBIS, onde a *posição* de um stakeholder é suportada por *argumentos*. Os argumentos podem ter origem numa ontologia.

O voto maioritário pode ajudar a medir a aproximação a um consenso, significando que pode ser considerado um mecanismo de ajuda à negociação.

O sistema de suporte ao SQFD tem de:

- permitir ao grupo compreender e negociar as correlações pretendidas pelos diversos stakeholders;
- resolver os conflitos por consenso.

O esquema 3.8 resume a negociação de uma correlação. O valor é obtido por consenso, em que cada stakeholder pode justificar a sua posição com argumentos de uma ontologia. Os stakeholders podem ser influenciados na sua posição pelo valor escolhido pela maioria.

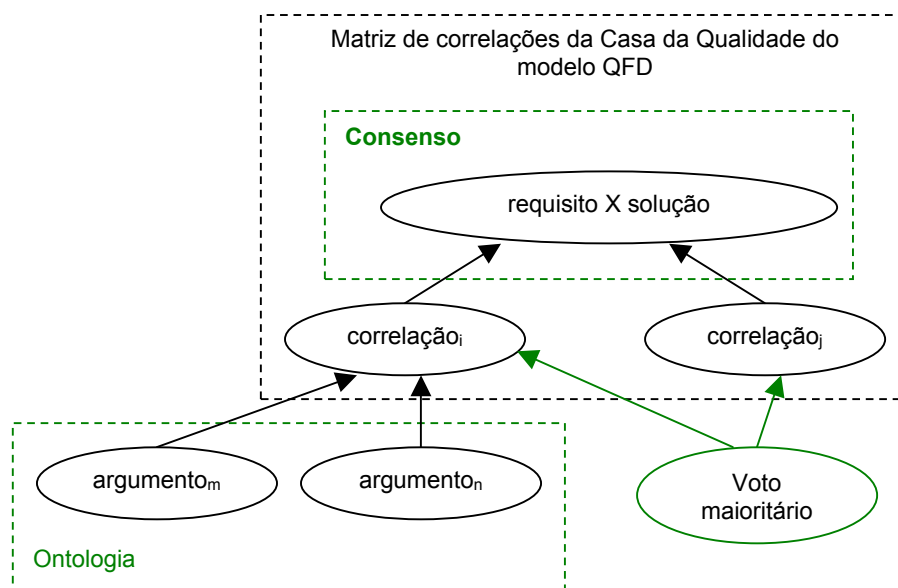


Figura 3.8. Esquema de suporte à negociação de correlações na Casa da Qualidade

Capítulo 4

Trabalhos relacionados

4.1. Contexto

A intervenção dos sistemas informáticos no processo de negociação pode ocorrer numa das seguintes situações [Kersten, 1987]:

- facilitar o processo;
- mediar activamente a negociação.

Baseado nesta amplitude e na teoria da negociação, deverá ser possível não só responder à questão *quando e como os diferentes sistemas de suporte à negociação são usados*, mas também permitir comparar e avaliar os diferentes sistemas [Kersten, 1987]. Quando um stakeholder conhece o papel que o sistema representa e as restrições inerentes, pode fazer uma escolha racional da ferramenta de suporte:

- pode escolher uma ferramenta restritiva e de mediação por causa do tempo limitado que tem para tomar uma decisão;
- se pretender entender o problema em profundidade ou se preparar para uma negociação (simulações), poderá preferir uma ferramenta de suporte menos restritiva.

4.2. Suporte básico dos sistemas de negociação

Os sistemas descritos nesta secção referem-se à infra-estrutura básica que permite suportar negociação.

4.2.1 Sistemas de correio electrónico

Nos sistemas de correio electrónico os stakeholders trocam mensagens acerca dos requisitos, podendo fazer parte de um grupo de difusão onde todas as mensagens são recebidas por todos. Para se manter uma sequência lógica das mensagens, um stakeholder terá de ser responsável pela gestão de mensagens de modo a garantir coerência.

4.2.2 Placares informativos

Os placares informativos são sistemas que permitem aos stakeholders enviar e receber mensagens de interesse geral, tendo normalmente associados uma *questão* a partir da qual se inicia a discussão. Podem estar associados a um sistema de correio electrónico e permitem a focalização da discussão num item [Bulletinboards, 2004]. Apesar de exigir menor gestão, é necessário acompanhamento para que a discussão seja encurtada.

4.2.3 Bases de dados partilhadas

As bases de dados partilhadas permitem:

- vários grupos de trabalho acederem simultaneamente à informação sem provocarem inconsistência nos requisitos, já que os mecanismos da base de dados garantem a integridade dos dados;
- obter informação de forma flexível recorrendo a facilidades de pesquisa;

- estabelecer ligações entre os requisitos e informação complementar.

Pretende-se por vezes armazenar não só os pontos a negociar e os resultados, mas também o processo, o que dá origem a uma base de dados de histórico da negociação. Este registo pode ser usado *à posteriori* para estudar estratégias adoptadas e simular diferentes estratégias [Chaudhury et al., 1991].

As bases de dados relacionais foram desenhadas de forma a armazenarem grandes quantidades de registos do mesmo tipo com o mínimo de ligações possível entre eles, enquanto as base de dados de requisitos, com bastante menos registos, possuem registos com muitas ligações a outros tipos de dados (documentos, ficheiros de texto, outros requisitos). Se bem que as base de dados relacionais suportem este tipo de ligações, podem revelar-se ineficientes para um numero considerável de requisitos. As bases de dados orientadas a objectos tem uma estrutura que se adequa melhor à gestão de requisitos. Permitem diferentes tipos de dados (objectos) e a gestão das ligações entre si é de fácil compreensão [Behm, 2001].

4.3. Tipologias de sistemas de suporte à negociação

4.3.1 Sistemas de Suporte à Negociação (NSS)

Um Sistema de Suporte à Negociação (Negotiation Support System, NSS) é um sistema informático que suporta negociação entre stakeholders. Os NSS tradicionais não têm o objectivo de disponibilizar suporte à decisão, apenas mostrando os detalhes das ofertas praticadas anteriormente pelos stakeholders.

É importante a forma como os dados são representados de modo a que os stakeholders possam construir ofertas melhores.

O principal objectivo dos NSS é dotar os stakeholders de um guia que lhes permita entender se o processo está ou não perto da fase final de negociação [Belluci & Zleznikov, 1998].

Os NSS tentam conciliar as necessidades de negociação e as capacidades tecnológicas de modo a:

- gerir a complexidade do ambiente de negociação;
- guiar as acções competitivas ou colaborativas;
- reduzir o risco de negociações emotivas.

4.3.2 Sistemas de Suporte à Decisão em Grupo (GDSS)

Um Sistema de Suporte à Decisão em Grupo (Group Decision Support System, GDSS) é um sistema informático que facilita a solução de problemas não estruturados por um conjunto de stakeholders que trabalham juntos como uma equipa.

O principal objectivo dos GDSS é o de aumentar a eficácia das decisões em grupo através da partilha de informação entre os stakeholders. Este objectivo pode ser alcançado por [Karacapilidis & Papadias, 1997a]:

- remover impedimentos à comunicação;
- providenciar técnicas para estruturar a análise da decisão;
- direccionar sistematicamente a forma, a duração ou os conteúdos de uma discussão.

Karacapilidis & Papadias (1997a) referem que a decisão em grupo se confronta com alguns constangimentos:

- a decisão é feita através de debate e negociação entre um grupo de stakeholders. Os conflitos de interesse são inevitáveis, sendo necessário suporte ao consenso e ao compromisso. Cada stakeholder pode adoptar e sugerir a sua própria estratégia que preencha certos objectivos;
- mais informação pode originar a mudança de posição relativamente a uma questão. As posições dos stakeholders podem ser suportadas por argumentos;
- conhecimento comprovado pode não ser suficiente para a decisão. Juízos de valor, dependentes dos objectivos e funções dos stakeholders, revelam-se as questões mais críticas de resolver;
- sistema deve providenciar ferramentas aos stakeholders que lhes permitam participar numa discussão de forma natural, sem ser necessário dominar modelos matemáticos ou soluções informáticas complexas.

4.3.3 Comparação entre NSS e GDSS

Grunbacher et al. (2003) referem que a distinção entre NSS e GDSS como tipos específicos de sistemas de suporte para decisão conjunta é ambígua. Ambos os tipos de sistemas procuram suportar os stakeholders na gestão dos processos de decisão de forma interdependente. Contudo, a diferença pode ser notada quanto à natureza e intensidade do conflito. A dificuldade em distinguir entre os dois tipos está relacionado com a própria definição de negociação.

Em [Grunbacher et al., 2003] é referido que as tarefas de negociação são sempre caracterizadas por um conflito entre os interesses (ou valores) das partes envolvidas, enquanto que nas tarefas de decisão em grupo tal não será necessariamente o caso. Na decisão em grupo tem-se, na maior parte do tempo, uma atitude integrativa por parte dos membros do grupo, onde o conflito é resolvido pela partilha de informação. No entanto,

na negociação, apesar de existirem igualmente interesses comuns, existem normalmente atitudes distributivas que inviabilizam a representação do problema de forma conjunta. Por outro lado, alguns dos mecanismos de geração de decisões conjuntas em grupo, tal como o voto maioritário, não são apropriadas na negociação que necessita de consenso para atingir um acordo.

4.4. Sistemas de suporte ao processo de negociação

A pesquisa efectuada sobre sistemas de suporte à negociação possibilitou a colecção de exemplos que, no seu conjunto, deram origem a diversas ideias implementadas no sistema MEG. A explanação que se segue não pretende ser exaustiva quanto ao número de sistemas de suporte consultados, mas sim mostrar os exemplos que influenciaram directamente a condução da investigação dos objectivos do protótipo desenvolvido.

4.4.1 EasyWinWin

O sistema EasyWinWin [Grunbacher & Briggs, 2001] [Briggs & Grunbacher, 2002] é uma metodologia de negociação de requisitos, tendo por base a aproximação de negociação Win-Win [Horowitz et al., 1999] e suportada por um GSS promovendo o envolvimento e interacção dos stakeholders mais importantes. A aproximação Win-Win baseia-se na Teoria-W [Bohem et al., 1995], em que o princípio fundamental é que é condição necessária e suficiente para que um projecto tenha sucesso que todos os stakeholders sejam vencedores.

Usando o EasyWinWin, os stakeholders são guiados passo-a-passo através de uma negociação Win-Win onde os requisitos são obtidos, construídos, priorizados, percorrendo e resolvendo as questões de forma a serem obtidos acordos mutuamente satisfatórios.

As principais actividades do EasyWinWin incluem:

- identificação dos stakeholders mais importantes;
- obtenção das *condições vencedoras* dos stakeholders. Condições vencedoras são os requisitos que, a serem aceites, tornam os stakeholders satisfeitos;
- negociação de pacotes de soluções mutuamente satisfatórios. Entende-se por pacotes requisitos, arquitecturas, planos, componentes críticos, etc;
- monitorização baseada em parâmetros e controlo do equilíbrio Win-Win por todo o processo de desenvolvimento.

Os stakeholders expressam os seus objectivos como *condições vencedoras*: se todos concordam, as *condições vencedoras* tornam-se *acordos*. Quando os stakeholders não concordam, identificam a sua *condição vencedora* em conflito e registam os seus conflitos como *questões*. Neste caso, os stakeholders inventam *opções* para ganho mútuo. As *opções* passam a *acordos* quando todos os stakeholders concordam.

Grunbacher et al. (2003) explica que a ordenação das *condições vencedoras* é feita com base em dois critérios:

- *valor*: representando o grau de importância para o negócio referente à implementação desse requisito;
- *praticabilidade*: representando a facilidade de implementação a nível tecnológico, social, político ou económico.

Grunbacher et al. (2003) referem ainda que normalmente os clientes e os utilizadores não se pronunciam quanto à *praticabilidade* e os programadores quanto ao *valor*.

O contexto da negociação de requisitos é definido pelo modelo em espiral de desenvolvimento de software [Bohem et al., 1995].

4.4.2 MEDIATOR

O MEDIATOR [Jarke et al., 1987] é um sistema de suporte à negociação (NSS), em que as negociações são feitas à base da busca de consenso através da troca de informação e, quando o consenso é incompleto, por compromisso.

O sistema dá suporte a um facilitador, que por sua vez auxilia um conjunto de stakeholders a construir uma representação conjunta do problema (em alguns contextos a ferramenta pode suportar directamente os utilizadores sem um facilitador). A representação é feita quer graficamente quer em forma de matriz. Em cada estado da negociação a representação mostra o nível de consenso / conflito entre os stakeholders. A evolução da representação do problema pode ser descrita como uma procura de consenso através da partilha dos *pontos de vista* individuais que constitui a troca de informação. O objectivo será encontrar um ponto de intersecção entre os diferentes conjuntos formados pelos *pontos de vista*, como solução do conflito.

O processo de negociação é realizado em duas fases:

- 1) *integração dos pontos de vista* onde o facilitador procura uma representação conjunta do problema;
- 2) negociação propriamente dita.

Na primeira fase, integra os pontos de vista individuais de cada stakeholder numa representação de grupo. Note-se que cada ponto de vista individual apenas é conhecido pelo facilitador e por esse stakeholder. Na segunda fase trabalha-se apenas dentro do ponto de vista de grupo.

4.4.3 HERMES

O sistema HERMES, desenvolvido por Karacapilidis & Papadias (1997), tem como objectivo principal desenvolver um sistema *activo* que constantemente actualize o

estado do discurso. O sistema não somente captura a memória organizacional, mas também ajuda os utilizadores durante o processo de decisão, actuando como um assistente ou conselheiro dado que facilita a comunicação e recomenda soluções, mas deixando a decisão final das decisões para os stakeholders [Karacapilidis et al., 1997].

Possui um suporte à argumentação que tem a sua origem no modelo IBIS e mostra os resultados na forma de Fórum de Discussão. O sistema HERMES recomenda ainda uma escolha de entre as várias opções possíveis. Esta recomendação activa é baseada no contexto do discurso e obtida a partir da atribuição de pesos que vão sendo actualizados à medida da importância dada a uma posição de um stakeholder em detrimento de outro.

O consenso é conseguido através de um processo colaborativo, ou seja, baseado na compreensão das várias posições dos stakeholders para a resolução do problema, interesses competitivos, prioridades e constrangimentos.

4.4.4 VIRTUAL QFD

O VIRTUAL QFD [Herzwurm & Schockert, 1999] é constituído por um sistema baseado em páginas Web. Na página principal é possível aceder às funcionalidades da aplicação a partir de um esquema baseado no processo QFD:

- 1) painel de discussão, onde são debatidos os requisitos dos stakeholders que pretendem o sistema e as soluções propostas pelos stakeholders que o desenvolvem, de uma forma semelhante a placares informativos;
- 2) tabela da *Voz do Cliente* e tabela da *Voz do Engenheiro* onde são estruturados os itens divulgados através do painel de discussão, sendo posteriormente categorizados;

- 3) painel de avaliação onde os stakeholders que pretendem o sistema podem priorizar os seus requisitos recorrendo ao *Analytical Hierarchy Process* (AHP);
- 4) Casa da Qualidade, onde os stakeholders inserem os valores de correlação. A importância dos itens da Voz do Engenheiro é calculada automaticamente com base nos valores do painel de avaliação e nos valores de correlação.

Herzwurm & Schockert (1999) apontam como aspectos negativos do sistema:

- o esforço do preenchimento da Casa da Qualidade e a necessidade constante de verificação das alterações nas correlações;
- as razões das alterações não serem compreendidas dado que os utilizadores não necessitam de as comentar.

Um ponto interessante do VIRTUAL QFD é o facto de permitir aos stakeholders inserirem e atribuírem pesos aos requisitos antes das reuniões QFD. As reuniões servirão muito mais para consolidar opiniões do que para gerar ideias. Na prática, muito do trabalho realizado durante as reuniões pode ser feito antes, o que torna o processo muito mais eficiente.

As conclusões essenciais apresentadas do uso do sistema são as vantagens inerentes às aplicações Web e a aplicação do QFD para diferentes grupos ter de levar em conta não só o peso dos requisitos mas também os diferentes valores de correlação para uma mesma célula da Casa da Qualidade: simplesmente calcular a média de valores introduzidos pelos vários grupos para a célula conduz a resultados distorcidos e esconde as necessidades contrárias entre os diversos grupos.

4.4.5 CO-DECIDE

O CO-DECIDE [Jacobs & Kethers, 1994][Gebhardt et al., 1997] consiste numa ferramenta que implementa a Casa da Qualidade do QFD numa forma editável pelos stakeholders. Cada stakeholder manipula a Casa da Qualidade a nível local e os resultados são sincronizados entre os diversos stakeholders, onde os conflitos de valores são evidenciados na matriz. Cada stakeholder tem acesso à informação da Casa da Qualidade segundo os seus privilégios (diferentes vistas do problema).

Todas as correlações, requisitos e características técnicas são o resultado de um processo de decisão. Normalmente as decisões são realizadas depois de uma discussão.

As posições que se encontram em conflito, os argumentos que suportam ou se opõem às posições são gravadas de modo a tornar as decisões, isto é, as entradas na Casa da Qualidade *referenciáveis*, sendo as discussões apoiadas por um editor de argumentação orientado ao IBIS.

4.4.6 Pontos considerados no protótipo a desenvolver

O sistema EasyWinWin foi um dos primeiros sistemas estudados e segue uma aproximação dominante nos sistemas de suporte à negociação de requisitos, ou seja, assume que os stakeholders seguem sempre atitudes integrativas: os objectivos / preferências individuais podem ser agregados numa representação do problema de forma conjunta e os conflitos são resolvidos pela facilitação do acesso à informação e à troca de conhecimento.

No entanto, em muitas situações é mais razoável assumir que os stakeholders têm conflitos de interesse e necessidades que não podem ser resolvidas simplesmente pela troca de informação. Nestes casos, uma estratégia interessante será aumentar a aproximação integrativa com métodos da teoria da negociação [Grunbacher et al., 2003]. Na verdade, a negociação de requisitos inclui tanto elementos integrativos como

distributivos, de modo a aumentar a sua cota dos valores disputados: um mero processo de facilitação que encoraje a troca aberta de informação será com certeza insuficiente para se alcançar sucesso. Por outro lado, factores socio-emocionais por exemplo, não garantem que os stakeholders tenham atitudes genuinamente integrativas.

A metodologia EasyWinWin suporta a obtenção gradual das preferências dos stakeholders mas apresenta falhas no suporte completo à negociação. Por exemplo, Grunbacher et al. (2003) referem que o sistema não providencia nenhum suporte na resolução de conflitos e na determinação de soluções alternativas.

Pretende-se portanto um sistema que não só suporte a troca de informação numa forma positiva quanto ao resultado do conflito, mas também permita o compromisso quando o consenso é incompleto. O objectivo será encontrar um ponto de intersecção entre os diversos pontos de vista dos stakeholders, à semelhança do MEDIATOR. O sistema MEG mostra igualmente o nível de consenso / conflito entre os stakeholders em cada estado da negociação, e quando encontra um ponto de intercepção (único) propõe esse valor para ser sujeito a consenso. Por outro lado, segue um processo de negociação semelhante ao descrito para o MEDIATOR de forma faseada:

- integra os interesses pessoais numa representação em grupo, em que a informação pessoal de cada stakeholder é resguardada dos restantes;
- a negociação é feita dentro da representação do grupo onde se procura o consenso.

A representação em grupo é conseguida, tal como no HERMES, na forma de Fórum de Discussão, onde o sistema não só captura a memória organizacional mas também recomenda soluções, deixando no entanto a decisão final para os stakeholders (é necessário consenso). Esta recomendação é baseada no contexto das posições e comportamentos dos stakeholders.

Um aspecto importante na escolha da metodologia QFD foi o factor *linguagem* associado ao factor *percepção*. O modelo permite transformar os requisitos expressos em linguagem natural num conjunto de termos técnicos, permitindo desfazer quaisquer ambiguidades entre o que os stakeholders que pretendem o sistema escrevem e o que o foi interpretado pelos stakeholders que o desenvolvem.

Durante o estudo da metodologia QFD, deparou-se com o sistema VIRTUAL QFD que deu origem a uma abstracção da negociação de requisitos confinada às células da Casa da Qualidade. Sendo o valor das correlações um dos factores que permitem a priorização dos requisitos / soluções, e havendo a possibilidade de vários valores de correlação para cada célula, exige-se que exista negociação dos valores de correlação para ser achado um valor *único* em cada célula (só assim é possível obter o nível de importância entre as possíveis soluções).

Por outro lado, a necessidade de compreender as razões das posições assumidas pelos stakeholders sugere que exista também um suporte à argumentação. Este suporte pode ser baseado no modelo IBIS numa aproximação igualmente explorada pelo CO-DECIDE.

O VIRTUAL QFD influenciou igualmente o sistema MEG na forma como pode ser usado antes das reuniões QFD, ou seja, muito do trabalho poderá ser feito antes das reuniões. Inclusivamente os conflitos deveriam ser resolvidos antes, e só os que realmente não conseguiam um consenso é que deveriam ser resolvidos na reunião QFD.

A tabela 4.1 apresenta um resumo dos principais aspectos que influenciaram o desenho do sistema MEG.

Tabela 4.1 – Resumo dos principais aspectos que influenciaram o desenho do sistema MEG

Sistema	Características
EasyWinWin	Abordagem Win-Win Priorização de requisitos
MEDIATOR	Consenso (ponto de intersecção) Só o sistema conhece tudo de todos Integra as vistas individuais e apresenta resultado global
HERMES	Fórum de discussão Valor proposto <i>activo</i>
VIRTUAL QFD	Valores das células em conflito: <i>necessidade de negociação entre valores inseridos para obtenção de um valor único</i> Suporte à inserção de valores antes das reuniões QFD
CO-DECIDE	Suporte à argumentação tipo IBIS associada ao QFD

4.5. Conclusões

A falta de compreensão dos requisitos expressos pelos stakeholders está na base de muitos problemas, sendo os sistemas electrónicos uma forma de permitir um acordo mais rápido:

- maior comunicação entre as partes geograficamente espalhadas e redução de custos de deslocação;
- a personalidade dos stakeholders não influencia os requisitos da mesma forma que aconteceria pessoalmente;
- não ser necessário uma resposta imediata permite uma tomada de consciência face às implicações de um requisito.

Se os stakeholders seguirem uma estratégia integrativa, partilham opiniões honestas sem quaisquer reservas. No entanto poderá ser adoptada uma relação baseada numa estratégia distribuída, o que origina que os stakeholders encubram os seus alvos e necessidades reais de forma a maximizar os seus próprios interesses. A utilização de sistemas de negociação na vida real tem sido condicionada pela sua dependência no comportamento cooperativo por parte dos stakeholders e na negligência dos processos de dar e receber no processo de negociação. A negociação de requisitos vai portanto muito para além do simplesmente se facilitar o processo e a troca de conhecimento.

As ferramentas de suporte à decisão multicritério revelam-se promissoras, dado que não só permitem comparar as diversas *questões* dos vários stakeholders, como também permitem a convergência dos interesses em conflito, e ainda considerar parâmetros como satisfação, análise de custos, etc. [Grunbacher et al., 2003].

Capítulo 5

Solução proposta

5.1. Contexto

O protótipo MEG é simultaneamente um sistema de suporte à negociação (NSS) e um sistema de suporte à decisão em grupo (GDSS), assente no modelo de desenvolvimento de software SQFD e fomentando uma perspectiva de negociação integrativa [Ramires & Antunes, 2004].

A abordagem apresentada vai descrever apenas o contexto da Casa da Qualidade para Software (software-HoQ), dado que o processo a aplicar nas restantes matrizes é análogo.

5.2. Negociação no modelo SQFD

No SQFD são usadas matrizes em que se comparam os requisitos com as soluções na forma de correlações (*corr*). Para estas correlações são possíveis os valores *nenhum*, *fraco*, *médio* e *forte* que equivalem aos valores numéricos 0, 1, 3 e 9 respectivamente. As correlações identificam a medida de interesse da correspondência entre a solução apresentada para o requisito proposto. No conjunto dos stakeholders S podem existir vários valores possíveis para *corr* o que pode originar conflito de interesses, exigindo portanto negociação (figura 5.1).

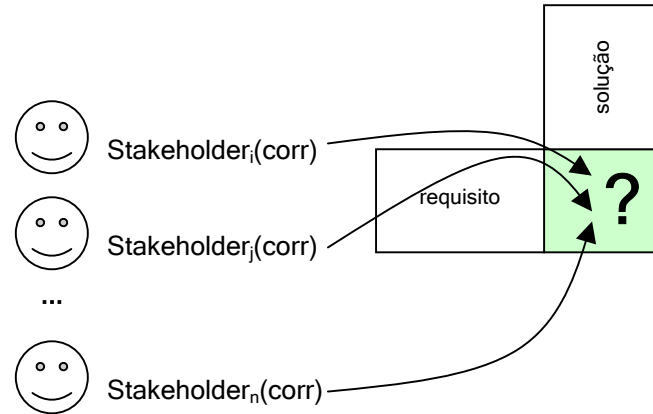


Figura 5.1. Conflito entre stakeholders para um valor de correlação

A resolução de conflitos de *corr* afecta directamente a priorização dos requisitos no SQFD. Pretende-se um sistema que auxilie a resolução do conflito de *corr* entre stakeholders e que obtenha um valor para *corr* por consenso. O sistema propõe ainda um valor baseado em voto maioritário e comportamento distributivo dos stakeholders.

Assim, seja TAB o conjunto de tabelas do modelo SQFD e swHoQ constituído por requisitos *r* e características *c* ligados por correlações *corr*, tem-se que:

$$\text{swHoQ} = \{ \{ r, c, \text{corr} \} \}, \text{swHoQ} \in \text{TAB} \text{ e } \text{TAB} \in \text{SQFD}$$

onde

$$\text{corr} = \{ \text{NENHUM} \mid \text{FRACO} \mid \text{MEDIO} \mid \text{FORTE} \}$$

Seja também

S_{TODOS} o conjunto de todos os stakeholders presentes no sistema,

S_{CELULA} o conjunto de todos os stakeholders envolvidos na discussão de uma célula CÉLULA tal que $S_{CELULA} \subset S_{TODOS}$,

S_i e S_j stakeholders $\in S_{CELULA}$ com S_i diferente de S_j .

Assume-se que o processo se desenrola numa CÉLULA de swHoQ, em que CÉLULA pode estar num estado de Equilíbrio E ou de Negociação T:

$$CÉLULA = \{ \{ E | T \}, \{ corr, S_{CELULA} \} \}$$

Na abordagem exposta no capítulo 2, o conjunto de atitudes possíveis na negociação é composto por:

$$ATITUDE = \{ \text{COMPETIÇÃO} | \text{ACOMODAÇÃO} | \text{INACÇÃO} | \\ \text{COLABORAÇÃO} | \text{COMPROMISSO} \}$$

O conjunto ATITUDE pode ainda ser dividido em atitudes Distributivas e Integrativas conforme a figura 5.2:

$$ATITUDE_{DIST} = \{ \text{COMPETIÇÃO} | \text{INACÇÃO}_{DIST} \}$$

$$ATITUDE_{INTE} = \{ \text{COLABORAÇÃO} | \text{ACOMODAÇÃO} | \\ \text{COMPROMISSO} | \text{INACÇÃO}_{INTE} \}$$

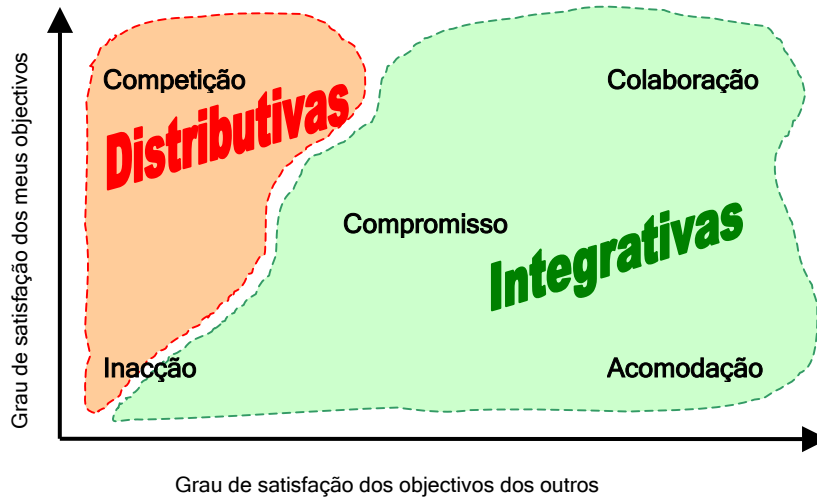


Figura 5.2. Atitudes distributivas e integrativas

A separação da atitude INACÇÃO deriva da definição encontrada em [Lewicki & Litterer, 1985]:

- INACÇÃO_{INTE} significa que o stakeholder se retira da situação de conflito, ou seja, retira todos os seus valores de CÉLULA permitindo à outra parte fazer o que quiser. Neste caso, é seguida uma estratégia integrativa, aplicável normalmente para os casos de negociação considerados insignificantes;
- INACÇÃO_{DIST} cria uma dependência na negociação por si próprio, frustrando os opositores. Neste caso, é seguida uma estratégia distributiva, onde se pretende, por exemplo, demorar a negociação para benefício próprio¹.

Relativamente ao suporte à argumentação (SA), a solução proposta apresenta

¹ Este comportamento poderia ser conseguido igualmente numa fase de confirmação quanto ao consenso, fazendo demorar a negociação; não seria possível, porém, distinguir entre o desejar *de facto* demorar o processo ou simplesmente ainda não ter respondido.

$$SA = \{ \text{questão}, \{ \text{posição}(S_i), \{ \text{argumento}(S_i) \} \} \}$$

Em que questão²:

$$\text{questão} = \{ \text{QUESTÃO} \mid \text{QUESTÃO_PROPOSTA} \mid \\ \text{QUESTÃO_FIRME_STAKEHOLDER} \}$$

Entende-se

- QUESTÃO como o valor para *corr* que origina a discussão³;
- QUESTÃO_PROPOSTA como o valor proposto para *corr* pelo sistema face ao contexto da negociação;
- QUESTÃO_FIRME_STAKEHOLDER como o valor proposto para *corr* por um stakeholder com uma atitude explícita de competição.

Os valores de cada stakeholder não são conhecidos dos restantes – apenas o sistema conhece os valores de todos.

O sistema facilita a tomada de posição de um stakeholder perante a *questão*: em vez de o stakeholder indicar explicitamente a sua *posição* relativamente à *questão*, o sistema assume implicitamente a *posição* do stakeholder, seja relativamente à QUESTÃO, seja relativamente à QUESTÃO_PROPOSTA baseada nos valores de todos. Assim:

$$\text{posição}(S_i) = \{ \text{NENHUMA} \mid \text{CONTRA} \mid \text{A_FAVOR} \}$$

² Questão pode ser representada na forma de pergunta “Qual o valor da correlação que relaciona o requisito com a solução?” conforme o ponto 3.4 do capítulo 3, ou no próprio valor de correlação que lhe serve de resposta.

³ Para uma célula vazia será o primeiro valor *x* inserido por um stakeholder *S_i*. Para uma célula preenchida será o resultado de um processo de negociação (valor único).

Entende-se ONTOLOGIA como o conjunto de palavras reservadas que fazem sentido num determinado contexto da negociação. Na solução proposta os argumentos dos stakeholders são expressos por palavras reservadas ARG:

$$\text{argumento}(S_i) = \{ \text{ARG} \} \in \text{ONTOLOGIA}$$

As relações entre entidades de SA são:

- questão TEM posições
- posição SUPORTADO_POR argumentos

A opção de um stakeholder face a uma solução apresentada pelo sistema para a resolução do conflito via consenso será:

$$\text{opção} = \{ \text{CONCORDO} \mid \text{NÃO_CONCORDO} \}$$

A solução proposta visa suportar a estratégia exposta no item Estratégia Integrativa do capítulo 2:

- Esconde a identidade dos stakeholders quando têm comportamentos integrativos. Ao esconder a identidade, os stakeholders atacam o problema em vez de se atacarem mutuamente, indo ao encontro dos pontos a) e b);
- Propõe um valor (QUESTÃO_PROPOSTA) que baseado em voto maioritário e no comportamento dos stakeholders, cria uma opção para ganho mútuo, de forma a promover justiça. Este comportamento expressa os pontos c) e d).

A estratégia integrativa é condicionada pelo grau de desconfiança das técnicas usadas pelos outros, originando que muitos ganhos conjuntos não sejam alcançados por permanecerem escondidos. Procura-se combater este condicionalismo ocultando

igualmente os valores das correlações possíveis de cada stakeholder. Em princípio, os stakeholders colocam os valores que pretendem de facto e os valores que podem tolerar, mesmo com um nível de satisfação inferior, de modo a viabilizar o consenso, mas sem se expor demasiado “abrindo o jogo”.

Por outro lado, o ponto a) do item Estratégia Distributiva do capítulo 2, refere a negociação firme como uma posição dura e inflexível onde não existe preocupação pelas necessidades dos outros. Este comportamento é possível na solução proposta, mas penalizado de modo a procurar conduzir os stakeholders para uma estratégia integrativa – uma atitude $INACÇÃO_{DIST}$ é igualmente penalizada pela mesma razão. O sistema proposto favorece o ponto b) e c), suporta os pontos a) e d) e inibe os pontos e), f) e g) do item Estratégia Distributiva.

5.3. Máquina de estados

É apresentado a máquina de estados para a solução proposta em que se descrevem os estados possíveis para CÉLULA – estado de Equilíbrio E ou de Negociação T.

Sendo $CÉLULA = \{ \{ E | T \}, \{ corr, S_{CÉLULA} \} \}$, os estados E e T podem ser definidos como

$$E = \{ a, b \} \text{ e } T = \{ n, o, p, x, z \}$$

A passagem entre estados é representada na figura 5.3 seguido de uma descrição sucinta da máquina de estados e de uma descrição do estado de Negociação T com o suporte à argumentação.

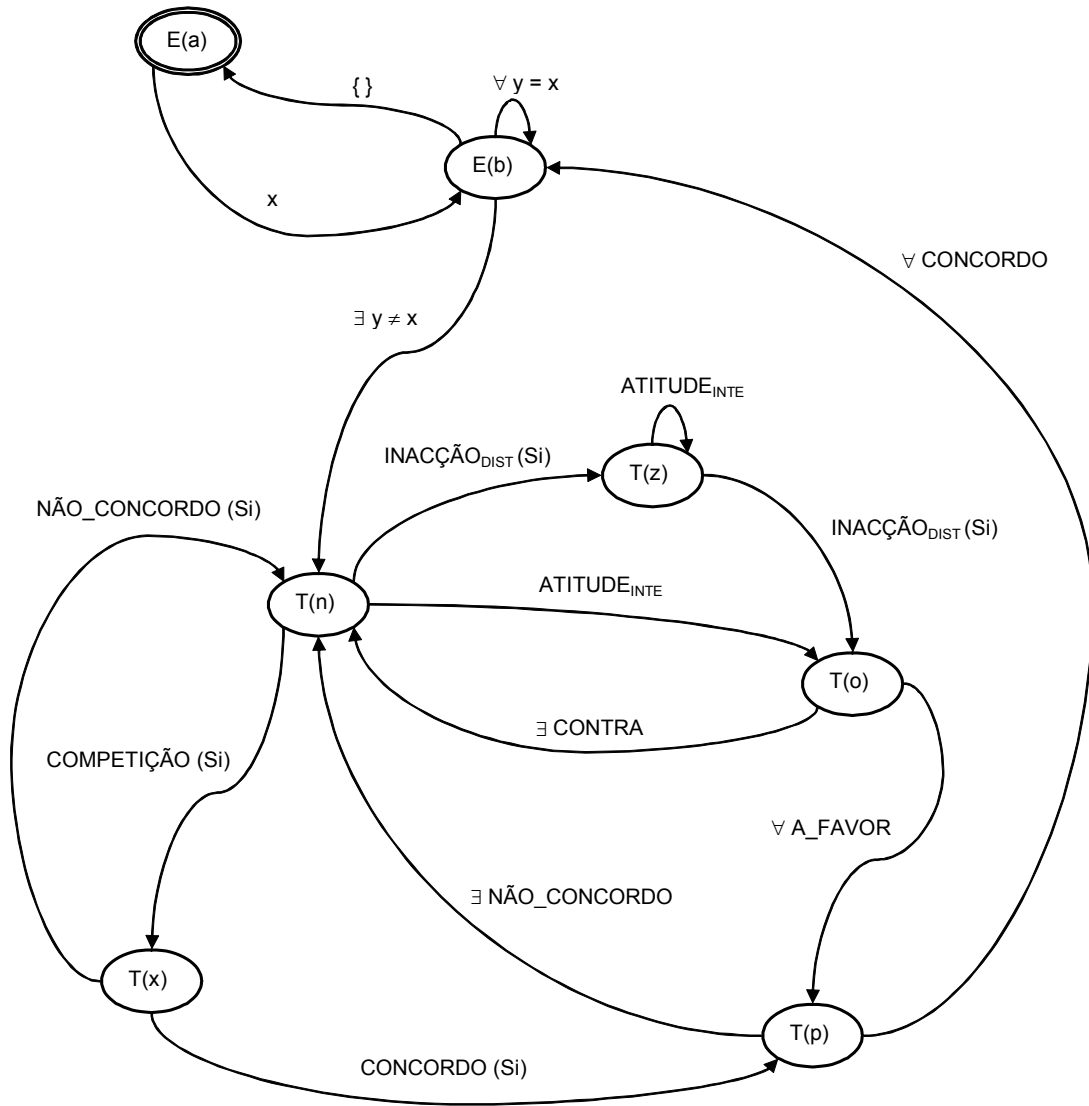


Figura 5.3 Máquina de Estados

Inicialmente, uma célula encontra-se vazia – estado $E(a)$ ⁴. Por inserção de um valor x na célula, passa-se a um estado de célula preenchida $E(b)$ ⁵. A inserção de novos valores y iguais ao valor x presente na célula não alteram o estado $E(b)$. Se forem retirados todos os valores da célula, passa-se ao estado $E(a)$.

Se no estado $E(b)$ ⁶ um stakeholder S_j insere um valor y diferente de um valor x que se encontra na célula inserido por um outro stakeholder S_i , passa-se a um estado de conflito $T(n)$ ⁷. Senão, mantém-se o estado $E(b)$ ⁸.

Para resolver o conflito é necessário que os stakeholders tomem uma atitude – Integrativa ou Distributiva.

Numa atitude integrativa $ATITUDE_{INTE}$ os stakeholders aumentam o intervalo de valores aceitáveis, ainda que com níveis de satisfação diferentes para cada valor do intervalo. Esta alteração provoca a passagem ao estado $T(o)$ em que se verifica se existe um ponto de intersecção entre os novos intervalos de valores. Se existe (ponto de intersecção em que todos são a favor, $\forall A_FAVOR$), passa-se ao estado $T(p)$. Caso contrário (se existe pelo menos uma posição contra, $\exists CONTRA$) retorna-se ao estado $T(n)$.

No estado $T(p)$, verifica-se se é possível o consenso entre stakeholders. Se todos concordarem no ponto de intersecção encontrado ($\forall CONCORDO$), passa-se ao estado de equilíbrio $E(b)$. Caso exista alguém que discorde ($\exists N\tilde{A}O_CONCORDO$), retorna-se ao estado $T(n)$.

⁴ Esta situação pode ser descrita formalmente como $C\acute{E}LULA = \{ E(a), \{ \} \}$.

⁵ Ou seja, $E(a) \rightarrow x \rightarrow E(b)$.

⁶ $C\acute{E}LULA = \{ E(b), \{ \{ corr(x), S_i \} \mid \{ \{ corr(x), S_i \}, \{ corr(y), S_j \}, x = y \} \} \}$.

⁷ $E(b) \rightarrow y \neq x \rightarrow T(n)$.

⁸ $E(b) \rightarrow y = x \rightarrow E(b)$.

No entanto, num estado de conflito $T(n)$, os stakeholders podem tomar uma atitude distributiva $ATITUDE_{DIST}$. Uma atitude $COMPETIÇÃO$ explícita por um stakeholder S_i ($COMPETIÇÃO(S_i)$) provoca a passagem ao estado $T(x)$. No estado $T(x)$ o stakeholder S_i autor da atitude de competição tem a possibilidade de confirmar a sua atitude. Se pretender continuar ($CONCORDO(S_i)$) passa ao estado $T(p)$, onde o valor proposto por S_i é sujeito a consenso. Se não pretende continuar ($NÃO_CONCORDO(S_i)$) retorna-se ao estado $T(n)$. O autor S_i de $COMPETIÇÃO(S_i)$ não necessita de confirmar a sua opção no estado $T(p)$.

Por outro lado, se um stakeholder S_i tiver uma atitude $INACCAO_{DIST}$ ($INACCAO_{DIST}(S_i)$) a partir de $T(n)$, passa-se ao estado $T(z)$ tornando impossível a passagem ao estado $T(p)$. A passagem ao processo de consenso $T(p)$ é deste modo bloqueado, o que inviabiliza a possibilidade da obtenção de um valor para a célula enquanto se encontrar no estado $T(z)$. Apenas o autor S_i da atitude $INACCAO_{DIST}$ tem a possibilidade de voltar a colocar a célula no estado $T(o)$ – a partir de $T(z)$ se o autor S_i tiver uma atitude $INACCAO_{DIST}$ ($INACCAO_{DIST}(S_i)$) passa-se ao estado $T(o)$ onde se verifica se existe um ponto de intersecção entre os intervalos de valores eventualmente criados por $ATITUDE_{INTE}$ em $T(z)$.

Segue-se uma descrição de cada estado de negociação T acompanhado do suporte à argumentação SA.

Estado $T(n)$

Foi inserido em CÉLULA um valor de *corr* y por um stakeholder S_j diferente do valor presente x inserido por S_i :

$$CÉLULA = \{ T(n), \{ \{ corr(x), S_i \}, \{ corr(y), S_j \}, x \neq y \} \}$$

Dado que existem pelo menos dois valores diferentes em CÉLULA, estamos num estado de conflito. O sistema gera automaticamente o seguinte SA:

$$\begin{aligned}
 SA = \{ & \text{QUESTÃO,} \\
 & \{ \text{posição}(S_i) = A_FAVOR, \{ \} \} \\
 & \{ \text{posição}(S_j) = CONTRA, \{ \} \} \\
 & \}
 \end{aligned}$$

Seja o valor x o valor colocado em CÉLULA por S_i . Sendo o valor y de S_j diferente do valor x , temos que a *posição* de S_j é CONTRA, e a *posição* de S_i será A_FAVOR ao valor de CÉLULA. Neste estado inicial da negociação não é necessário que os stakeholders insiram *argumentos*.

Para resolver o conflito é necessário os stakeholders tomarem uma atitude:

$$\begin{aligned}
 T(n) & \rightarrow \text{ATITUDE}_{\text{INTE}} \rightarrow T(o) \\
 T(n) & \rightarrow \text{COMPETIÇÃO} \rightarrow T(x), \quad \text{para } S_i \\
 T(n) & \rightarrow \text{INACÇÃO}_{\text{DIST}} \rightarrow T(z), \quad \text{para } S_i
 \end{aligned}$$

Estado T(o)

Um stakeholder S_i alterou os valores de *corr* numa atitude Integrativa. Se existe um ponto de intersecção entre os intervalos definidos por cada stakeholder que origina uma *posição* A_FAVOR para a *questão* em $S_{\text{CÉLULA}}$ passa-se ao estado T(p).

Se não existe um ponto de intersecção, o sistema tenta propor um valor para *questão* (QUESTÃO_PROPOSTA) baseado em voto maioritário e no contexto das atitudes dos stakeholders. O processo será o seguinte:

Por cada valor escolhido, é atribuído um peso $w = 1$.

Se o stakeholder tiver tido n atitudes COMPETIÇÃO a partir de $T(n)$, é penalizado $n \cdot 10^{-3}$ vezes ou seja $w = 1 - (n \cdot 10^{-3})$.

Se o stakeholder tiver tido n atitudes INACÇÃO_{DIST}, é penalizado $2n \cdot 10^{-3}$ vezes ou seja $w = 1 - (2n \cdot 10^{-3})$.

Por exemplo, para as condições:

$\text{corr}(S_i) = 3$, com uma atitude INACÇÃO_{DIST} verificada anteriormente;

$\text{corr}(S_j) = 9$

Tem-se que:

$w_i(3) = 1 - (2 \cdot 1 \cdot 0,001) = 0,998$

$w_j(9) = 1$

Calculando o máximo entre os pesos w vem que

$\text{MAX}(w_i, w_j) = 1 \Rightarrow \text{QUESTÃO_PROPOSTA} = 9$

SA será formado por:

QUESTÃO = 3

posição = A_FAVOR

posição = CONTRA

QUESTÃO_PROPOSTA = 9

posição = CONTRA

posição = A_FAVOR

Note-se que depois de ser obtida a QUESTÃO_PROPOSTA, é aplicado o mesmo critério das *posições* utilizado na QUESTÃO relativamente aos valores de cada stakeholder face à *questão* correspondente – Si passa a ter uma posição CONTRA o valor 9 dado que a sua escolha é o valor 3 e Sj uma posição A_FAVOR do valor 9.

Apesar de o sistema se basear principalmente no voto maioritário para propor um valor, não impõe que este seja o escolhido. Apenas tenta mostrar a situação global e orientar na decisão mais justa. Se na situação descrita o máximo $\text{MAX}(w_i, w_j)$ não fosse único, o sistema não poderia propor um valor.

Tem-se portanto

$T(o) \rightarrow A_FAVOR \rightarrow T(p),$ para S_{CELULA}

$T(o) \rightarrow T(n)$ caso contrário

Os stakeholders poderão ainda indicar os *argumentos* de forma a serem compreendidas as razões das *posições* assumidas.

Estado T(p)

O estado T(p) pode ser atingido se se verificarem as seguintes condições em SA:

$$\{ \text{QUESTÃO} \mid \text{QUESTÃO_PROPOSTA} \}$$

$$\text{posição} = \text{A_FAVOR}, \quad \text{para } S_{\text{CELULA}}$$

Ou

QUESTÃO_FIRME_STAKEHOLDER conseguida através de uma atitude COMPETIÇÃO originada em T(n).

SA mostra portanto o valor da questão que originou T(p), com o fim de ser aprovado por consenso e passar-se ao estado E(b).

Se todos têm como opção CONCORDO, passa-se ao estado de Equilíbrio E(b). Se existe um NÃO_CONCORDO, passa-se ao estado de Negociação T(n):

$$T(p) \rightarrow \forall \text{ CONCORDO} \rightarrow E(b)$$

$$T(p) \rightarrow \exists \text{ NÃO_CONCORDO} \rightarrow T(n)$$

Na solução proposta, a noção de consenso é aplicada ao grupo S_{CELULA} presente na discussão de uma questão de uma CÉLULA, significando que pode não corresponder ao conjunto S_{TODOS} . Se o valor de um stakeholder entrar em conflito com o valor acordado anteriormente por um grupo de stakeholders, o processo de negociação recomeça (não é necessário aqui que todos os elementos do grupo façam parte do consenso; apenas é necessário consenso entre os que fazem parte da discussão de uma célula, dado que se alguém não entra na discussão da célula em questão é porque não está interessado naquele valor).

Estado T(x)

Se um stakeholder S_i teve uma atitude COMPETIÇÃO explícita a partir do estado de negociação T(n), e não volta com a sua decisão atrás, passa-se para T(p). Caso contrário retorna a T(n):

$$T(x) \rightarrow \text{CONCORDO} \rightarrow T(p), \quad \text{para } S_i$$

$$T(x) \rightarrow \text{NÃO_CONCORDO} \rightarrow T(n), \quad \text{para } S_i$$

SA mostra apenas QUESTÃO_FIRME_STAKEHOLDER.

Estado T(z)

Se um stakeholder S_i teve uma atitude INACÇÃO_{DIST} a partir do estado de negociação T(n), torna impossível a passagem ao processo de consenso – apenas S_i têm a possibilidade de fazer sair CÉLULA do estado T(z). A passagem ao processo de consenso é assim *bloqueado* o que inviabiliza a possibilidade de obtenção de um valor para CÉLULA enquanto S_i mantiver CÉLULA no estado T(z). Seja S_i o autor da escolha INACÇÃO_{DIST}, tem-se que:

$$T(z) \rightarrow \text{ATITUDE}_{\text{INTE}} \rightarrow T(z)$$

$$T(z) \rightarrow \text{INACÇÃO}_{\text{DIST}} \rightarrow T(o), \quad \text{para } S_i$$

Só é possível mostrar em SA os valores QUESTÃO e QUESTÃO_PROPOSTA. QUESTÃO_FIRME_STAKEHOLDER não é mostrado por apenas serem permitidas atitudes integrativas no estado T(z).

5.4. Processo de negociação

A situação inicial de um processo de negociação é de conflito de interesses, acompanhado pela necessidade de cooperação. O objectivo das partes não é o de convencer a outra parte da correcção da sua opinião mas de se chegar a um acordo. Um compromisso implica a passagem de uma posição inicial para outra posição que vá ao encontro de um valor que satisfaça ambas as partes (mesmo que em níveis de satisfação diferentes). O processo de negociação (figura 5.4) consiste em 3 fases:

- fase de pré-negociação;
- fase de negociação;
- fase de pós-negociação.

5.4.1 Fase pré-negociação

A *fase de pré-negociação* apresenta o problema, isto é, prepara os pontos a negociar para a fase seguinte.

Descrição do problema

Corresponde ao estado E(b): cada stakeholder insere um valor de entre { 0, 1, 3, 9 } sem necessidade de apresentação de argumentos. Pretende-se obter apenas as posições iniciais dos stakeholders para uma visualização do problema. A inserção de um valor dá origem à passagem ao estado T(n).

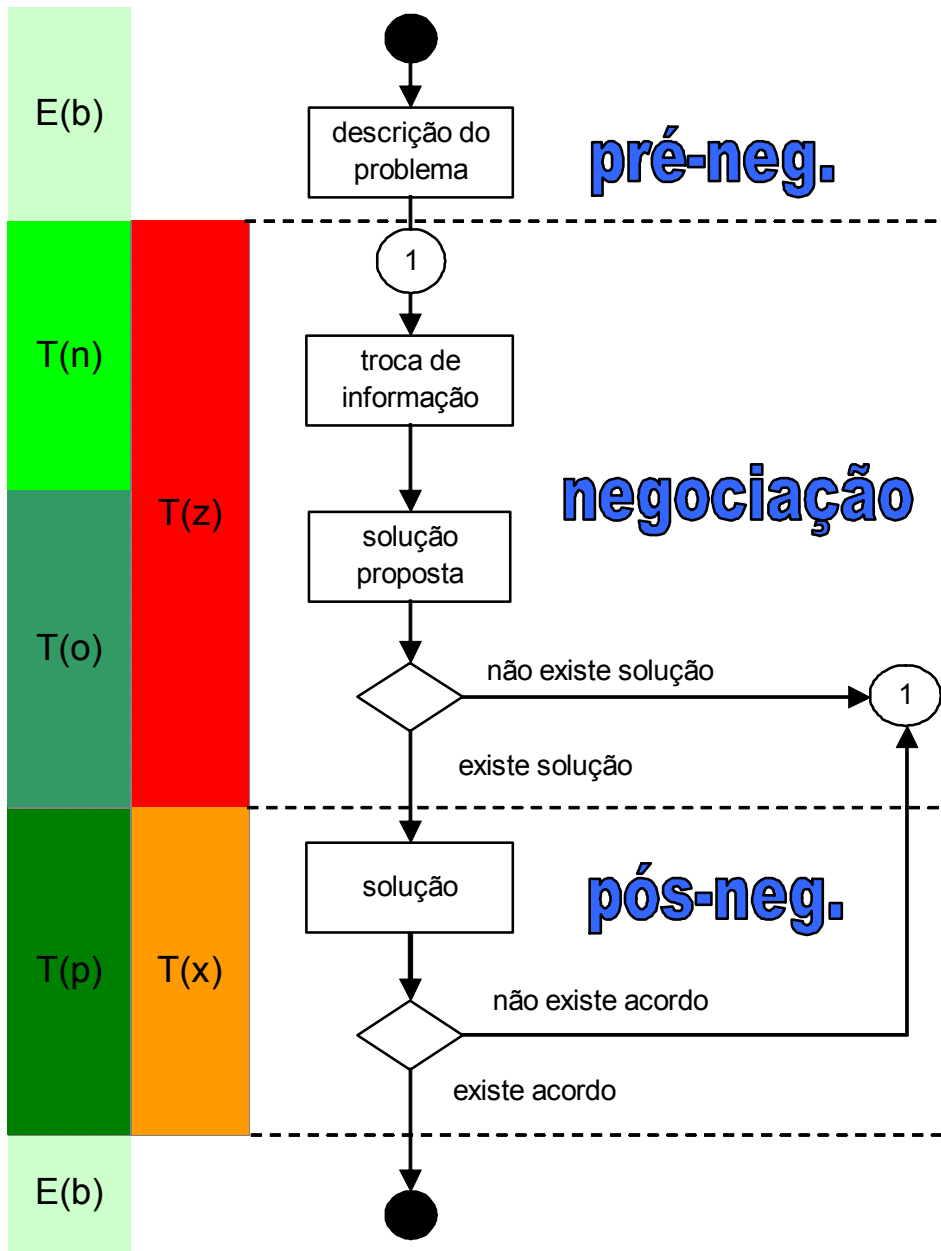


Figura 5.4 - Processo de negociação

5.4.2 Fase de negociação

Na *fase de negociação* são trocados valores e argumentos entre stakeholders. Se se chega a uma solução (*existe solução* na figura 5.4), é necessário obter o consenso dos stakeholders envolvidos na discussão (caso contrário continua a troca de valores / argumentos).

Troca de informação

Corresponde ao estado T(n): pode-se inserir mais do que um valor, significando o alargamento de valores possíveis de modo a encontrar um ponto de intersecção de interesses. Esta etapa representa atitudes implícitas de COMPROMISSO, COLABORAÇÃO e ACOMODAÇÃO, consoante os valores tendam para o valor da *questão*. Nesta etapa é possível a inserção de argumentos que sustentam as *posições*. O alargamento de valores origina a passagem ao estado T(o).

Uma atitude explícita de COMPETIÇÃO remete ao estado T(x), ou seja, um valor proposto de forma forçada por um stakeholder para ser sujeito a aprovação. O processo de aceitação é análogo ao de T(p).

Por outro lado, uma atitude INACÇÃO_{DIST} impossibilita a passagem à *fase de pós-negociação*, ou seja, não permite que se chegue a consenso.

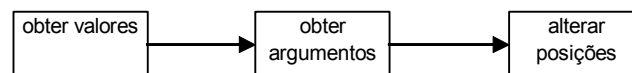


Figura 5.5 – Esquema de troca de informação

A alteração dos valores pode originar a alteração da *posição* de um stakeholder relativamente a uma *questão* (figura 5.5).

Solução proposta

Corresponde ao estado T(o): o sistema apresenta uma solução quando todas as posições dos stakeholders envolvidos forem A_FAVOR para QUESTÃO ou QUESTÃO_PROPOSTA. Se existe uma *posição* CONTRA, passa-se ao estado T(n).

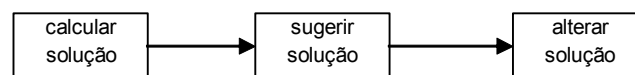


Figura 5.6 – Esquema de solução proposta

A solução proposta é conseguida a partir de novo cálculo do ponto de intersecção entre os valores dos stakeholders (figura 5.6).

5.4.3 Fase de pós-negociação

Se na *fase de pós-negociação* se chega a um consenso quanto à solução proposta (*existe acordo* da figura 5.4), o valor encontrado passa a ser a solução (caso contrário continua a troca de valores / argumentos). O consenso conduz a um estado de equilíbrio E(b) até ser introduzido novo valor que entre em conflito com a solução encontrada, e nesse caso temos a repetição do processo descrito.

Solução

Chegamos a uma *solução* em duas situações:

- todas as *posições* são A_FAVOR de um valor da QUESTÃO / QUESTÃO_PROPOSTA;

- existe um e um só S_i com uma atitude COMPETITIVA em $T(n)$, sendo o valor proposto o correspondente ao valor de S_i .

A solução apresentada é sujeita a consenso. Ou todos são a favor da solução a que se chegou (primeira situação) ou todos concordam com a solução proposta por um stakeholder (segunda situação). Se a solução é aprovada, reporta-se novamente ao estado E(b). Se não é aprovada e provém da primeira situação, depara-se um problema: retorna-se à situação que deu origem à passagem à *fase de pós-negociação*, e por conseguinte, *automaticamente* passa-se novamente a novo pedido de CONCORDO ou NÃO_CONCORDO aos stakeholders (entra-se em ciclo). Uma solução é guardar a situação em que se verificou a opção NÃO_CONCORDO e passar novamente à *fase de negociação*. Se a mesma situação se verificar, não passar à *fase de pós-negociação*.

Por exemplo, para uma QUESTÃO com todas as posições A_FAVOR, se existir um NÃO_CONCORDO volta-se à *fase de negociação*. A nível da SOLUÇÃO_PROPOSTA verifica-se se a solução é igual à última situação de NÃO_CONCORDO. Se for, não se passa para a *fase de pós-negociação* (apesar de poderem estar todos A_FAVOR, o que pode acontecer mesmo que mudem os valores de cada stakeholder). Salvaguarda-se a passagem à *fase de pós-negociação* para uma situação equivalente, de modo a evitar o entrar em ciclo, mas se *de facto* se pretender a passagem à *fase de pós-negociação* imediatamente a seguir para essa situação, tal não será possível – por exemplo, quem optou por NÃO_CONCORDO arrependeu-se. Apesar do comportamento descrito, optou-se por implementar essa solução.

5.5. Esboço do protótipo

Quem procede à inserção de um valor numa célula (quer a célula esteja vazia ou não), apenas pode inserir um e um só valor e ver o resultado na forma de *posições* relativamente a uma *questão*, podendo assumir uma atitude COMPETIÇÃO ou

ACOMODAÇÃO. A figura 5.7 mostra o esboço da interface referente à *fase de pré-negociação*. Este interface é disponibilizado a um stakeholder que não tenha valores inseridos em CÉLULA.

A *fase de pré-negociação* é caracterizada pela apresentação do problema. Essa apresentação é mostrada na caixa Current Situation que reflete a situação corrente do valor de CÉLULA. Se a CÉLULA se encontrar no estado E(a), não existe QUESTÃO (*Issue*); existe QUESTÃO se estiver no estado E(b); existe QUESTÃO e talvez QUESTÃO_PROPOSTA (*Issue_proposed*) se estiver no estado T(n) ou T(o).

Current Situation

issue
position

Value

0
 1
 3
 9

OK

Figura 5.7 – Esboço da interface da *fase de pré-negociação*

Na caixa Value é possível inserir um e um só valor por duas razões:

- se CÉLULA está no estado E(a), o valor de QUESTÃO têm de ser único;

- se CÉLULA está no estado E(b), T(n) ou T(o), um stakeholder toma uma atitude de COMPETIÇÃO ou ACOMODAÇÃO, inserindo respectivamente um valor diferente ou igual a QUESTÃO.

O botão OK permite a inserção do valor do stakeholder no sistema. Esta inserção permite ao stakeholder passar à *fase de negociação* da CÉLULA.

Na *fase de negociação*, importa perceber as razões das várias *posições* dos stakeholders, pelo que se possibilita a inserção de *argumentos*. A figura 5.8 mostra o esboço da interface referente à *fase de negociação*.

FIRM	LOCK	
<div style="border: 1px solid black; padding: 5px;"> <p>Current Situation</p> <p>issue position argument</p> <p>issue proposed position argument</p> </div>		
<div style="display: flex; justify-content: space-between;"> <div style="border: 1px solid black; padding: 5px; width: 45%;"> <p>Value</p> <p><input type="checkbox"/> 0</p> <p><input type="checkbox"/> 1</p> <p><input type="checkbox"/> 3</p> <p><input type="checkbox"/> 9</p> </div> <div style="border: 1px solid black; padding: 5px; width: 45%;"> <p>Arguments</p> <p>type</p> <p>my own arguments</p> <p><input type="checkbox"/> reason</p> <p>risk based arguments</p> <p><input type="checkbox"/> reason</p> </div> </div>		
<input type="button" value="OK"/>		

Figura 5.8 – Esboço da interface da *fase de negociação*

A caixa Current Situation mostra a situação corrente de forma semelhante à *fase de pré-negociação*, com o acréscimo da apresentação de *argumentos* que sustentam as *posições* dos stakeholders. A inserção de argumentos é realizada pela selecção de um conjunto de palavras reservadas (ontologia) a partir da caixa Arguments. O botão OK permite a inserção / alteração de valores do stakeholder no sistema.

As atitudes \in $ATITUDE_{INTE}$ são implícitas na alteração dos valores – nesta fase podem ser inseridos simultaneamente mais do que um valor a partir da caixa Value.

As atitudes $COMPETIÇÃO$ e $INACÇÃO_{DIST}$ são expressas respectivamente através de opções de menu FIRM significando que “mantenho a posição porque é absolutamente imprescindível que assim seja” e LOCK significando que “não quero negociar com ninguém” – inibe o processo de negociação. As atitudes \in $ATITUDE_{DIST}$ são implementadas como opções de menu de forma a serem menos acessíveis.

O sistema passa a $CÉLULA$ para a *fase de pós-negociação* quando um stakeholder escolhe a opção FIRM ou quando a *solução proposta* tem a posição A_FAVOR para $S_{CÉLULA}$.

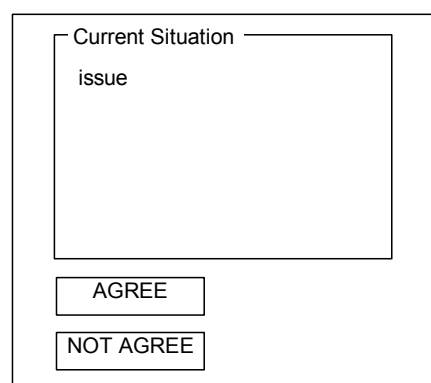


Figura 5.9 - Esboço da interface da *fase de pós-negociação*

Na fase de pós-negociação, se todos os stakeholders escolherem a opção CONCORDO, a CÉLULA passa a um estado de equilíbrio e é colocado esse valor na célula da matriz Casa da Qualidade para Software.

Na caixa Current Situation da figura 5.9, aparece apenas a *questão* que está sujeita a consenso.

Se a *questão* é do tipo QUESTÃO_FIRME_STAKEHOLDER, são revelados na caixa Current Situation o número de vezes de uso das opções FIRM e LOCK do autor da *questão*. Pretende-se fornecer o nível de atitudes distributivas de um stakeholder S_i que podem influenciar os outros stakeholders na aceitação ou não do valor proposto por S_i .

5.7. Conclusões

O sistema deverá ser usado prioritariamente antes das reuniões SQFD de modo a que durante as reuniões sejam resolvidos apenas os casos que:

- suscitam dúvidas. Neste caso, as *posições* dos stakeholders podem necessitar de mais esclarecimentos (mesmo as *posições* acompanhadas de *argumentos*).
- permanecem num impasse. Neste caso, a solução proposta é sempre rejeitada por alguém ou a CÉLULA não recupera de um estado $T(z)$.

Se a atitude predominante não for o COMPROMISSO, a utilização do sistema será frustrante pela quantidade de conflitos não resolvidos nas matrizes de SQFD – depende da atitude integrativa / distributiva dos stakeholders. Excesso de atitudes distributivas são mal interpretadas num ambiente que se promove atitudes integrativas: o número de vezes do uso da opção FIRM ou LOCK não só são revelados como também interferem na SOLUÇÃO_PROPOSTA.

As decisões são efectuadas num diálogo entre iguais: cada stakeholder tem o direito de fazer alterações dos valores pretendidos e prevenir alterações que julguem inaceitáveis. As *questões* são negociadas até que todos cheguem a um acordo.

A solução proposta visa:

- esconder a complexidade do processo de negociação na avaliação de requisitos de software;
- conduzir os stakeholders a assumirem atitudes integrativas sem limitar as suas opções;
- propor valores de consenso, requerendo no entanto que os stakeholders explicitamente concordem ou não com esses valores;
- manter uma reserva de privacidade necessária a um processo de negociação, dado que não são revelados os valores individuais que os stakeholders estão dispostos a aceitar.

Capítulo 6

Protótipo desenvolvido

6.1. Contexto

Estudos recentes revelaram que soluções de implementação de QFD baseadas em folhas de cálculo Microsoft EXCEL não produzem resultados inferiores a implementações elaboradas noutras linguagens de programação. Apesar de não haver uma única ferramenta que suporte todo o processo QFD, temos que, no geral, o nível de satisfação dos clientes ao usarem diversas ferramentas que implementam o processo é semelhante, incluindo soluções baseadas em EXCEL [Herzwurm et al., 1997].

Por outro lado, Ramesh & Mohan (2001) mostram que um grande impedimento à adoção de NSS e de GSS é que as aplicações são desenvolvidas de forma isolada e não integradas com ferramentas que são usadas diariamente no processo de trabalho.

Com base nestes pressupostos, adoptou-se uma solução que inclui o EXCEL como visualizador de dados de grupo e que integra uma ferramenta de negociação no suporte ao modelo SQFD.

6.2. Arquitectura

O sistema MEG é constituído basicamente por uma folha de cálculo EXCEL situada no computador pessoal de cada stakeholder e sincronizada a partir de um servidor (RTDSERVER) conforme mostra a figura 6.1. A interacção com cada stakeholder é estabelecida através de uma ferramenta de inserção de dados (MEGCLIENT) que por sua vez altera o repositório de dados a partir de um servidor (MEGSERVER).

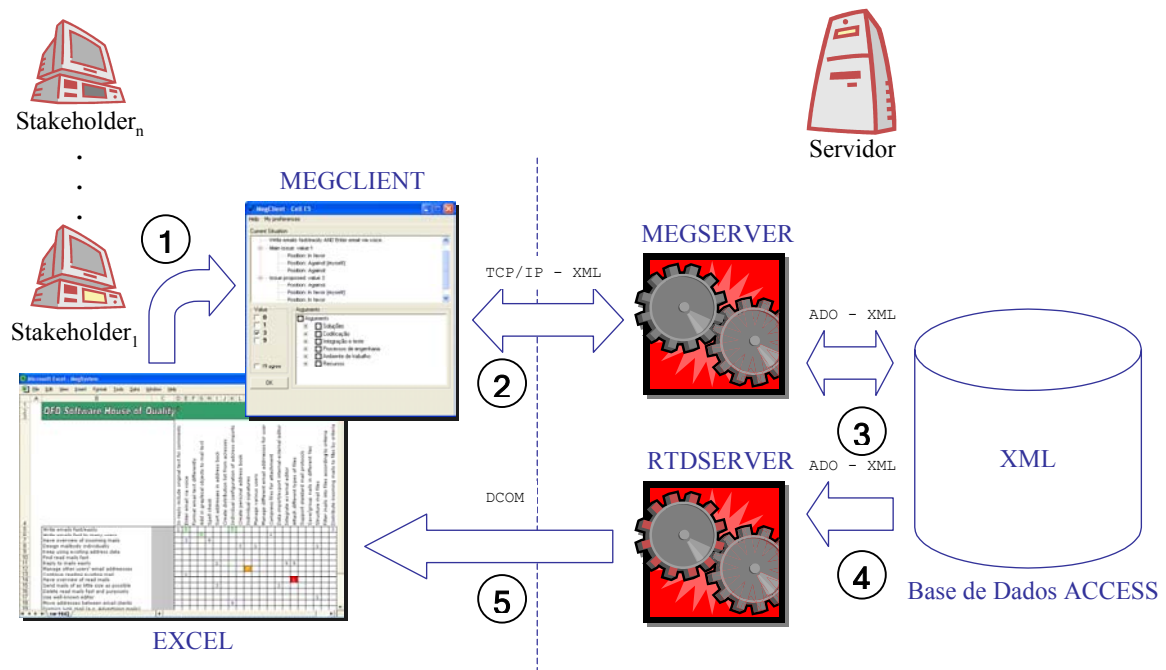


Figura 6.1. Arquitectura do sistema MEG

As folhas de EXCEL contém a Casa de Qualidade para Software. As correlações da matriz não podem ser alteradas directamente no EXCEL. A ferramenta MEGCLIENT invocada através de *duplo click* sobre a CÉLULA a alterar (caminho 1 da figura 6.1) comunica com o servidor MEGSERVER (caminho 2). O servidor MEGSERVER altera o repositório de dados XML (caminho 3) e actualiza os dados de MEGCLIENT

(caminho 2). As correlações das matrizes são sincronizadas através do servidor RTDSERVER que faz a leitura do repositório de dados XML (caminho 4) e comunica com as folhas de EXCEL (caminho 5).

São descritos de seguida os vários componentes do sistema MEG.

6.2.1 Microsoft EXCEL

É utilizado o Microsoft EXCEL 2002 por este disponibilizar uma nova funcionalidade que permite visualizar alterações feitas num repositório de dados em tempo real. Esta funcionalidade denominada *Real-Time Data* (RTD) [Getsch & Cornell, 2001][Cornell, 2001] permite ver as alterações que acontecem no repositório de dados sem intervenção dos stakeholders, algo pretendido em situações como a gestão de stocks ou actividades de *Warehouse*.

Esta nova funcionalidade é acedida através de formulas RTD colocadas em cada célula da matriz da folha de cálculo. Por esta razão, não é permitido no sistema MEG a manipulação directa das correlações na folha de cálculo – as folhas que implementam a Casa da Qualidade para Software são preparadas antes de os stakeholders as usarem. Encontrando-se protegidas, não só escondem o formato das funções, mas também não permitem a alteração das formulas, o que inviabilizaria a visualização dos dados. O formato normal da função RTD a colocar em cada célula das matrizes SQFD será:

```
=RTD(Classe, Servidor, ReferenciaCelula)
```

em que Classe representa a identificação da classe do objecto do servidor RTDSERVER (classe xRtd), Servidor representa o nome da máquina onde o servidor MEGSERVER está instalado e ReferenciaCelula representa as coordenadas da célula no formato ColunaLinha.

A formula RTD pode ser encapsulada de modo a facilitar a inserção num só local da Classe e Servidor. No MegSystem temos que cada célula contém a formula genérica:

```
=meg (ADDRESS (ROW () ; COLUMN () ; 4 )
```

definida num módulo de *Visual Basic for Applications* (VBA) da folha de cálculo da seguinte forma:

```
Function meg(c As String) As String
Dim cc As Variant
    cc = c
    meg = Excel.Application.WorksheetFunction.RTD(
        "MegServerRtd.xRtd", server_address, cc)
End Function
```

A representação de uma Casa de Qualidade para Software é apresentada na figura 6.2 tendo como base a tabela obtida a partir de Herzworm et al. (2002), em que se definem requisitos para um cliente de correio electrónico.

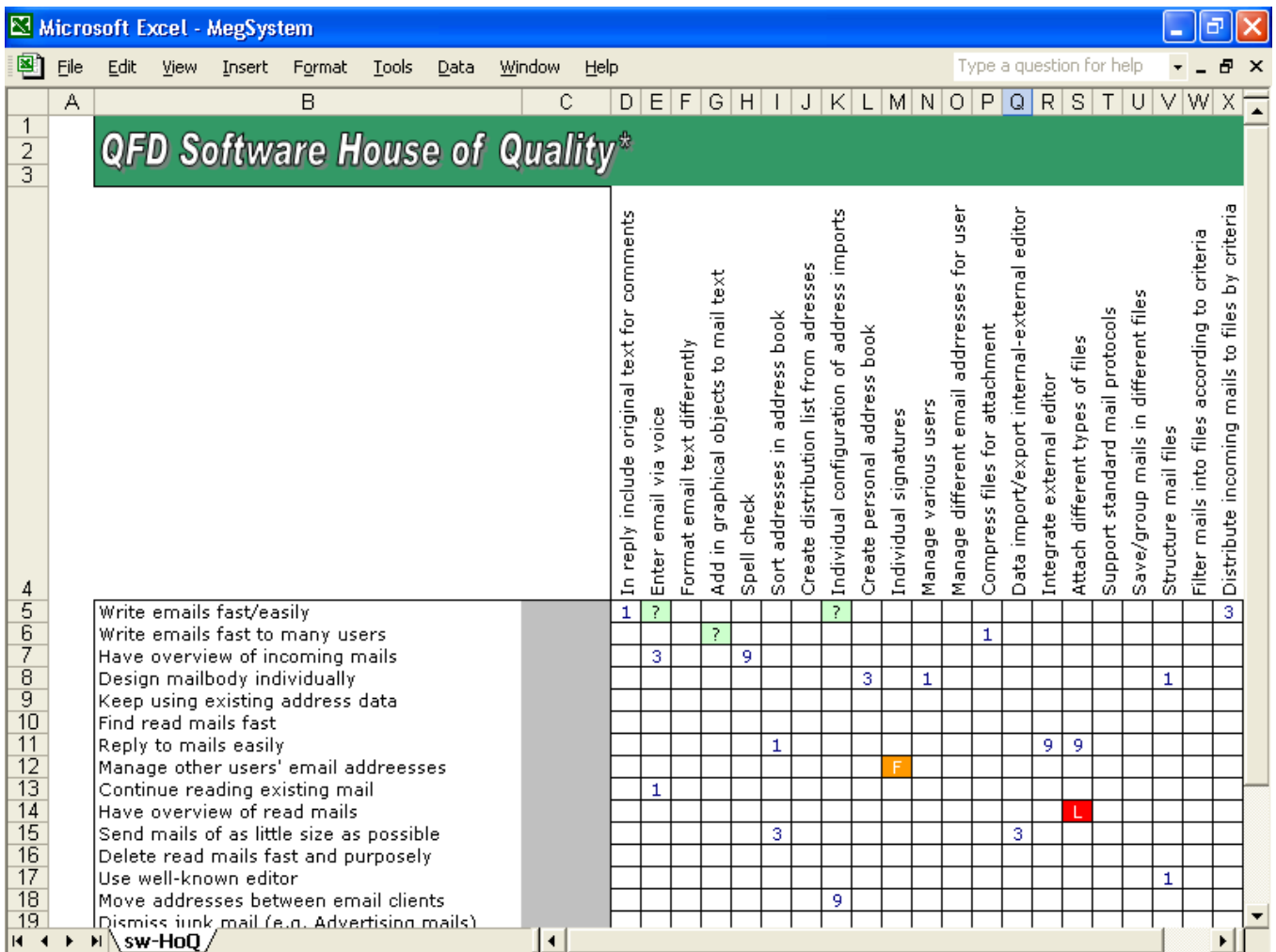
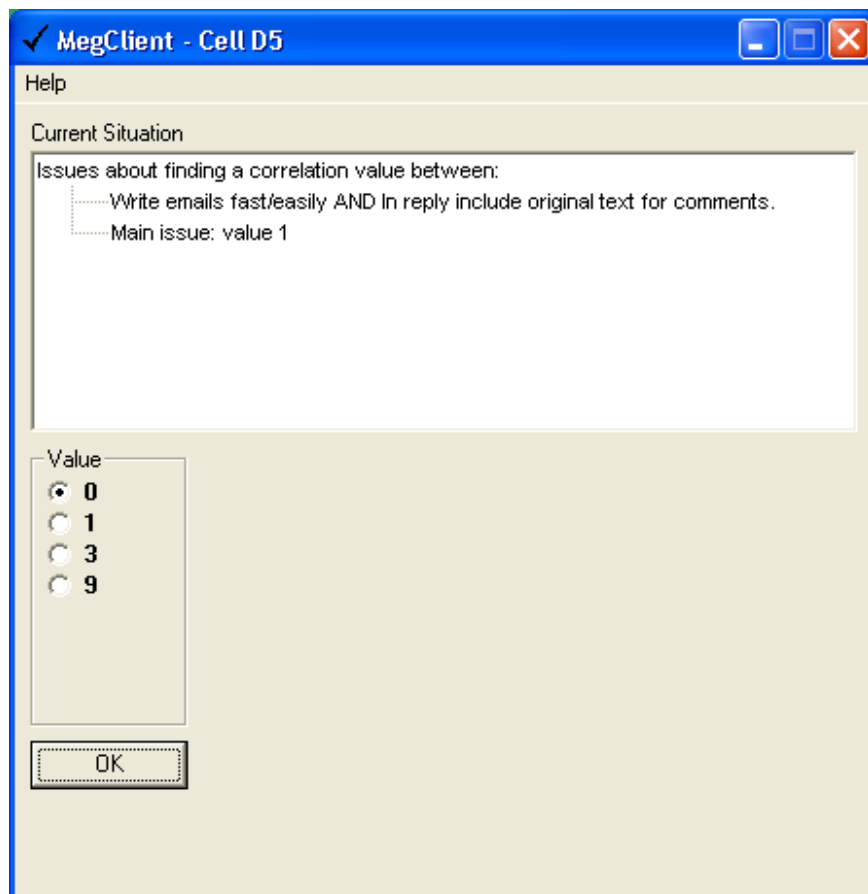


Figura 6.2 – Exemplo de Casa da Qualidade para Software (adapt. de [Herzwurm et al., 2002]) representada no sistema MEG

Os valores possíveis para uma CÉLULA são um valor de { 0, 1, 3, 9 } ou { ?, F, L } consoante a CÉLULA esteja respectivamente numa fase de negociação (?), um stakeholder tenha escolhido a opção FIRM (F) ou LOCK (L) do menu My preferences da figura 6.3.

6.2.2 MEGCLIENT

A ferramenta MEGCLIENT é invocada a partir do *duplo click* sobre uma CÉLULA referente às correlações da Casa da Qualidade (caminho 1 da figura 6.2), dando origem ao formulário mostrado na figura 6.3 para a *fase de pré-negociação*. Nesta fase, apenas é possível visualizar a situação corrente na caixa Current Situation, escolher um valor da caixa Value (botões do tipo *OptionButton* só permitem uma única escolha) e inserir o valor no sistema pela selecção do botão OK, conforme explicado no item 5 do capítulo 5.



The image shows a Windows-style dialog box titled "MegClient - Cell D5". The dialog has a blue title bar with standard window controls (minimize, maximize, close). Below the title bar, there is a "Help" section. The main content area is titled "Current Situation" and contains a text box with the following text: "Issues about finding a correlation value between: Write emails fast/easily AND In reply include original text for comments. Main issue: value 1". Below this text box, there is a "Value" section with four radio buttons labeled "0", "1", "3", and "9". The "0" radio button is selected. At the bottom of the dialog, there is an "OK" button.

Figura 6.3 – Formulário – fase de pré-negociação

No caso da figura 6.3, a CÉLULA da Casa da Qualidade para Software com as coordenadas D5 tinha o valor 1 inserido anteriormente por outro stakeholder. Este valor, que colocou a CÉLULA no estado E(b), originou a *questão* (Main issue) que poderá ser debatida com o stakeholder autor do formulário da figura 6.3. A caixa Current Situation mostra também o *requisito* “Write emails fast/easily” e a *solução* “In reply include original text for comments” da matriz da figura 6.2 para os quais se pretende um valor único de correlação, obtido por consenso.

O processo de alteração de uma CÉLULA consiste na ferramenta MEGCLIENT enviar uma mensagem em formato XML (comunicação via TCP/IP) para o servidor MEGSERVER (indicação 2 da figura 6.2.). O servidor MEGSERVER analisa o pedido, altera o repositório de dados (indicação 3) e devolve o resultado da alteração a cada ferramenta MEGCLIENT que estiver activa para aquela célula (sentido inverso da indicação 2). Por sua vez, o servidor RTDSERVER consulta o repositório (indicação 4) e actualiza as folhas de cálculo dos diversos stakeholders (indicação 5), inclusivamente a folha do stakeholder que fez o pedido de inserção (comunicação via DCOM).

O MEGCLIENT mostra a informação actualizada em tempo real, ou seja, qualquer alteração efectuada por um stakeholder para uma célula é reflectida na Current Situation de cada stakeholder que tiver activo para aquela mesma célula. Quando se acede à ferramenta, são reflectidas nas zonas Value e Arguments as escolhas tomadas anteriormente pelo stakeholder que o chamou.

Se na figura 6.3 o stakeholder seleccionar o valor 3 e a seguir o botão OK, a CÉLULA passa à *fase de negociação*.

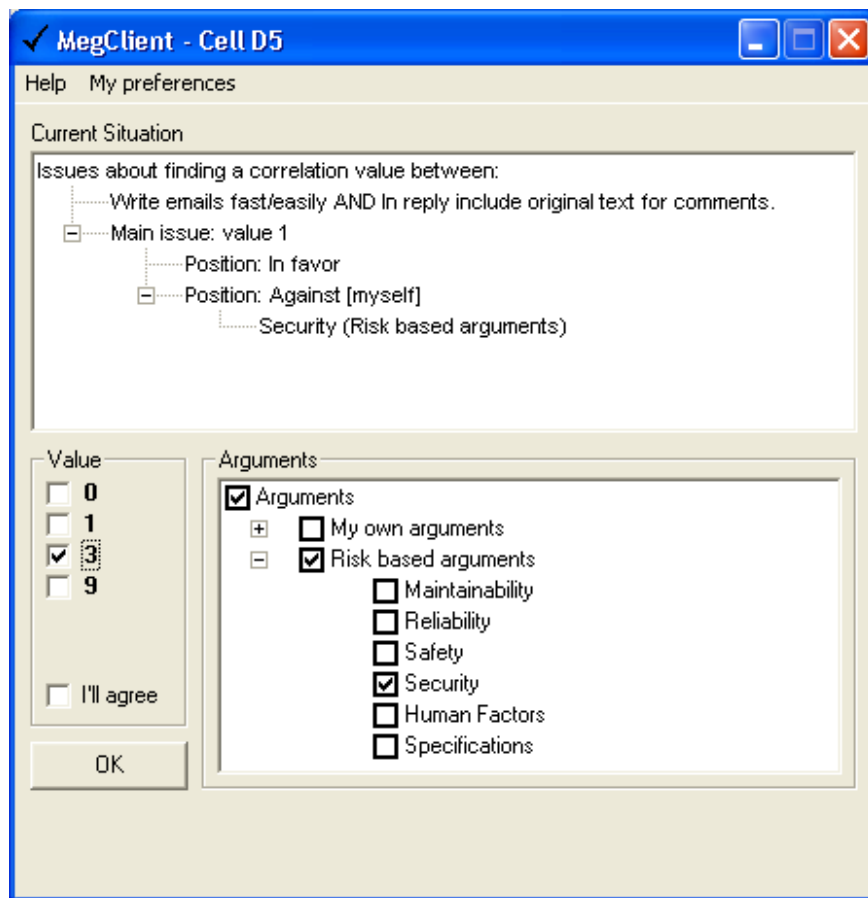


Figura 6.4 – Formulário – fase de negociação

A figura 6.4 mostra a *fase de negociação* para a CÉLULA D5. A caixa Value passou de botões do tipo *OptionButton* que só permitem uma única escolha para botões do tipo *CheckBox* que permitem escolha múltipla, com vista a permitir atitudes Integrativas; disponibilização da caixa Arguments onde é permitido seleccionar cada *argumento* em grupo ou individualmente; a opção de menu *My preferences* com as opções FIRM e LOCK (atitudes Distributivas) e uma opção I'll agree.

A caixa Current Situation expressa as *posições* (a favor e contra) dos stakeholders relativamente à *questão*. A *posição* de cada stakeholder é acompanhada pela indicação myself, visível apenas para o autor dessa *posição* e que ajuda cada stakeholder a situar a sua *posição* no processo de negociação. Estas *posições* são obtidas em função dos valores inseridos pelos stakeholders ($posição = F(\text{correlação})$), em que apenas são dadas a conhecer as *posições*. A relação entre a caixa Current Situation e a figura 3.8 do capítulo 3, é apresentada na figura 6.5.

Formulário da fase de negociação – figura 6.4

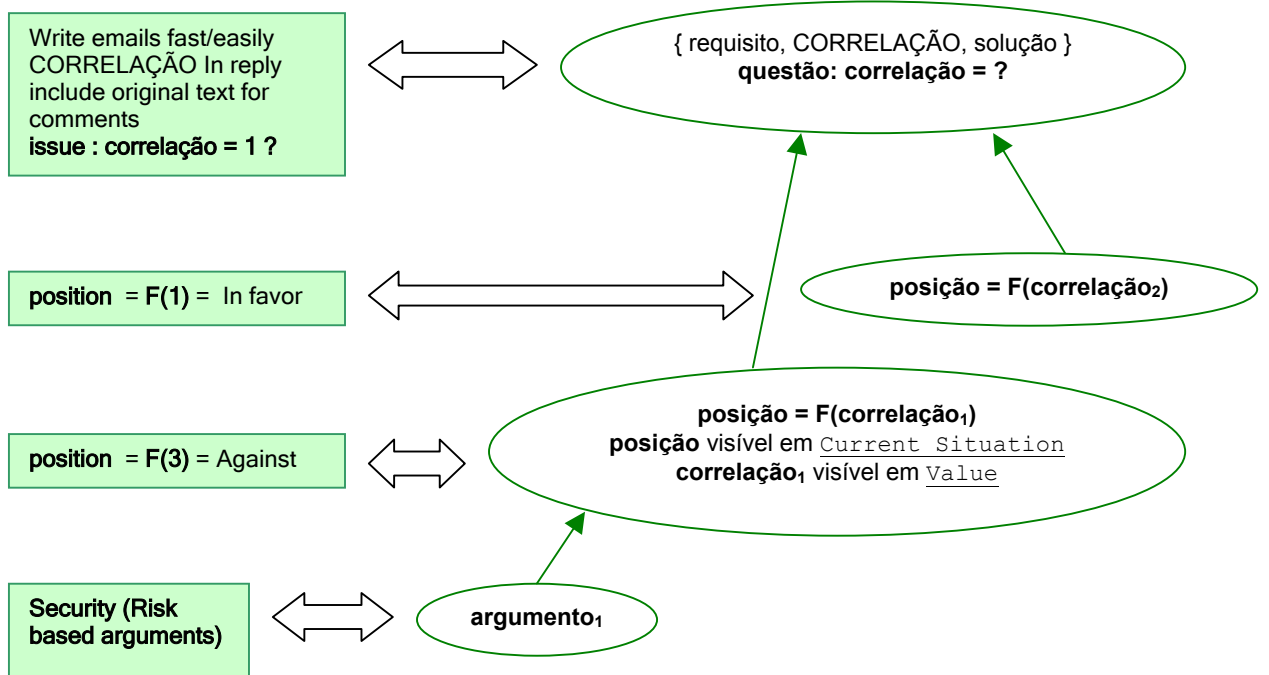


Figura 6.5 – Relação entre o formulário da figura 6.4 e a figura 3.8

Os argumentos apresentados na figura 6.4 são meramente exemplificativos. Esta questão é abordada com maior pormenor mais à frente no item referente ao repositório de dados.

A opção I'll agree seleccionada permite-nos dizer *à priori* que em caso de passagem à *fase de pós-negociação*, concordamos com os valores em negociação desde que sejam iguais aos seleccionados por nós em Value.

Dado que, por um lado, nem sempre é possível testar o sistema MEG em ambiente de rede na fase de desenvolvimento e, por outro, o EXCEL não permite ter de forma prática varias instâncias de uma folha de cálculo num mesmo computador, foi criada uma versão de teste. Nesta versão, o MEGCLIENT é executado directamente, sendo necessário fornecer as coordenadas da CÉLULA em que se pretende inserir um valor que entre em conflito com um já presente nessa célula e a identificação do stakeholder que se quer representar. Da mesma forma, o programa pode ser testado num computador que não tenha o EXCEL instalado; apenas necessitamos de clientes teste e do servidor.

A figura 6.6 mostra o formulário da *fase de pré-negociação* em versão de teste. A caixa Test version apresenta no campo Srv o nome da máquina onde está instalado o MEGSERVER, o nome do stakeholder que se quer assumir através da selecção do campo Stk e a CÉLULA correspondente através do campo Cell. Após a selecção dos campos Stk e Cell, selecciona-se o botão Connect para estabelecer a ligação com o servidor (este processo é feito de forma transparente para o stakeholder na versão normal do MEGCLIENT).

Na versão teste podem ser também consultadas as mensagens que são enviadas pelo servidor MEGSERVER na caixa Received From Server para efeitos de *debug*. Dado

que na versão teste não são inseridos requisitos e soluções, estes não aparecem na Current Situation.

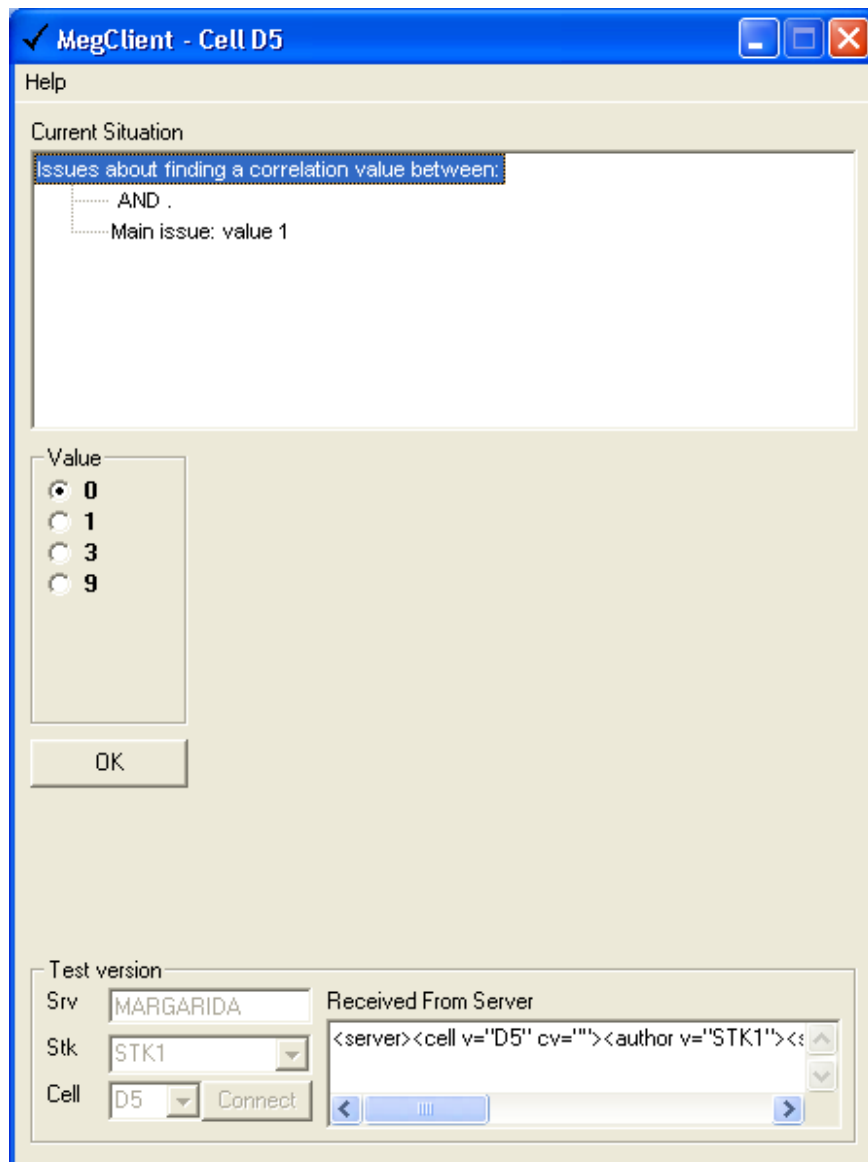


Figura 6.6 – Formulário VERSÃO TESTE – fase de pré-negociação

Se por exemplo, para a situação da figura 6.4, o stakeholder inserir o valor 1 que permite a passagem à *fase de pós-negociação* (foi conseguido um ponto de intersecção entre os conjuntos { 1 } e { 1, 3 }) fica-se numa situação semelhante à mostrada na figura 6.7. Se o stakeholder seleccionar AGREE relativamente à *questão* indicada na caixa Current Situation, e quando os stakeholders que pertencem ao conjunto S_{CELULA} tiverem seleccionado o botão AGREE, o conjunto S_{CELULA} é avisado através de uma caixa de diálogo (figura 6.9) de que se chegou a consenso quanto ao valor de CÉLULA. Após se premir o botão OK, o MEGCLIENT referente a essa célula é fechado e na Casa da Qualidade para Software, o sinal “?” é substituído pelo novo valor encontrado.



Figura 6.7 – Formulário – fase de pós-negociação

Se na situação da figura 6.4 o stakeholder tivesse inserido o valor 1 e seleccionado a opção I'll agree, não teria sido mostrado o formulário da *fase de pós-negociação*. Apesar de a CÉLULA se encontrar numa situação de obtenção de consenso, o formulário mantinha-se semelhante ao formulário da *fase de negociação* mas com a mensagem “Cell in agreement stage” como mostra a figura 6.8. Nesta situação, o botão OK encontra-se inactivo para não permitir a alteração de valores durante esta fase.

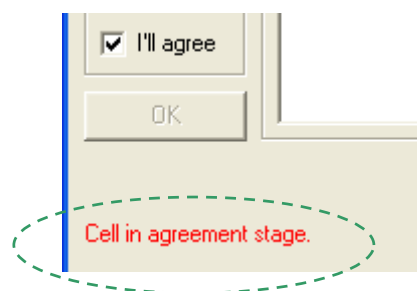


Figura 6.8 – Mensagem de *fase de pós-negociação* com I'll agree



Figura 6.9 – Mensagem em caso de consenso

Se um dos stakeholders optar por seleccionar NOT AGREE, o conjunto S_{CELULA} é avisado através de uma mensagem (figura 6.10) de que não foi possível chegar a consenso quanto ao valor de CÉLULA. O MEGCLIENT dessa célula passa a mostrar o formato correspondente à *fase de negociação* (figura 6.4) em que se encontrava antes de passar à fase de *pós-negociação* (figura 6.7).



Figura 6.10 – Mensagem em caso de falha de consenso

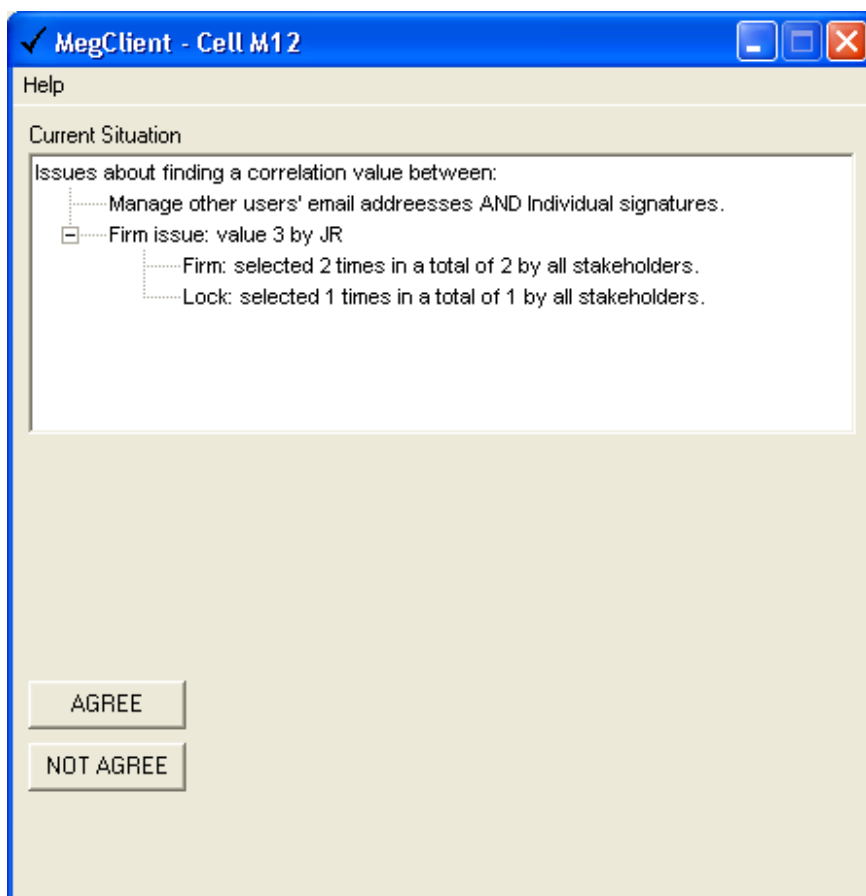


Figura 6.11 – Formulário – fase de pós-negociação com FIRM

A opção de menu My preferences / FIRM a partir de um formulário como o da figura 6.4 conduz à figura 6.11 (a figura mostra agora a situação para a CÉLULA M12).

A caixa Current Situation identifica não só o autor da opção FIRM, como também revela o número de vezes de uso da opção FIRM ou LOCK no número total em que essas opções são utilizadas, o que influencia o consenso quanto ao valor proposto pelo autor da opção FIRM – é diferente aceitar um valor proposto por um stakeholder que tem um grande número de atitudes distributivas relativamente a um outro que não tem esse tipo de atitudes (o excesso de atitudes distributivas é mal interpretado num ambiente que promove atitudes integrativas).

A opção My preferences / LOCK conduz a um formulário como o da figura 6.12 (a figura mostra a situação para a CÉLULA S14). O autor da opção LOCK é identificado numa mensagem de aviso (WARNING) na caixa Current Situation.

A opção LOCK provoca com que as opções de menu My preferences fiquem inactivas para todos os stakeholders que participam em CÉLULA com excepção do autor da opção que continua com acesso à opção LOCK para que consiga retirar a escolha efectuada.

É possível continuar na *fase de negociação* alterando valores e argumentos. No entanto, se for encontrada uma *solução* (todas as *posições* estão A FAVOR de uma *questão*), não é possível obter o consenso (a opção LOCK inibe a passagem à *fase de pós-negociação*) mantendo-se o mesmo formulário. Se for retirada a opção de LOCK e todas as posições forem A FAVOR de uma questão, a CÉLULA passa automaticamente à *fase de pós-negociação*.

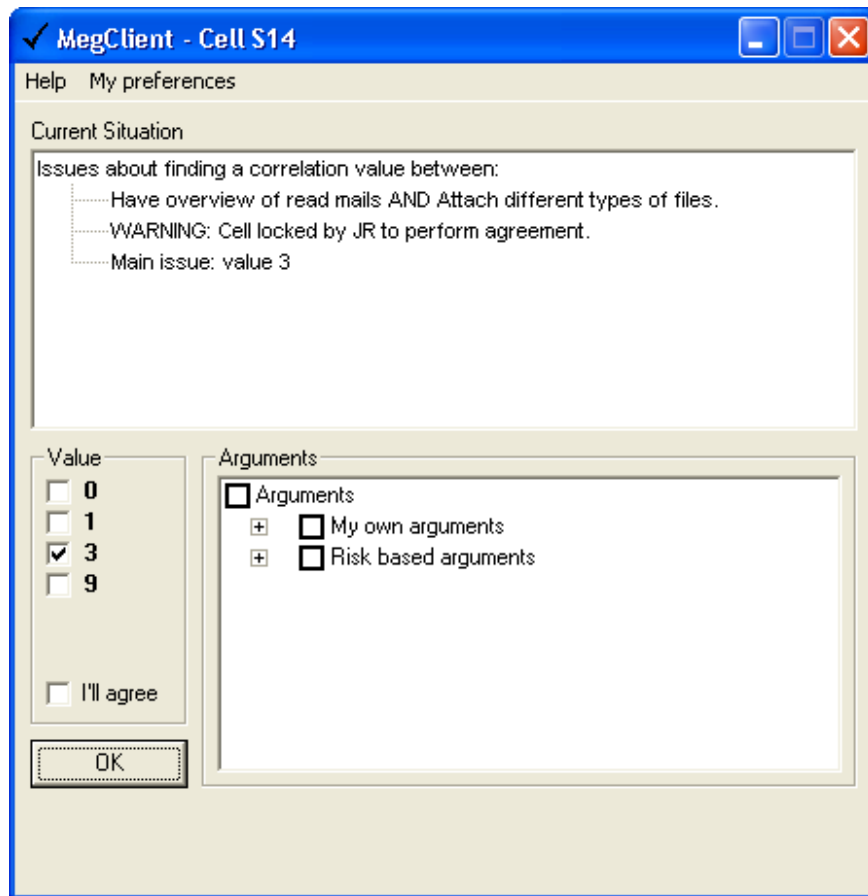


Figura 6.12 – Formulário – fase de negociação com LOCK

A ferramenta MEGCLIENT foi elaborada em VISUALBASIC 6.0 Enterprise Edition com SP5. A ferramenta envia e recebe mensagens XML para e do servidor MEGSERVER através de TCP/IP. O MEGCLIENT tem como base a parte *cliente* de [Planetsourcecode, 2003].

6.2.3 MEGSERVER

O servidor MEGSERVER é responsável pela gestão do repositório de dados e dos pedidos elaborados pelos clientes MEGCLIENT dos diversos stakeholders.

A figura 6.13 apresenta o formulário do servidor com a caixa Received que contém as mensagens XML recebidas dos clientes MEGCLIENT referentes a cada CÉLULA de cada stakeholder. O botão Reset insere um novo registo com *tags* XML inicializados a zero / branco nas tabelas ALOCTAB e MEGREP do repositório de dados.

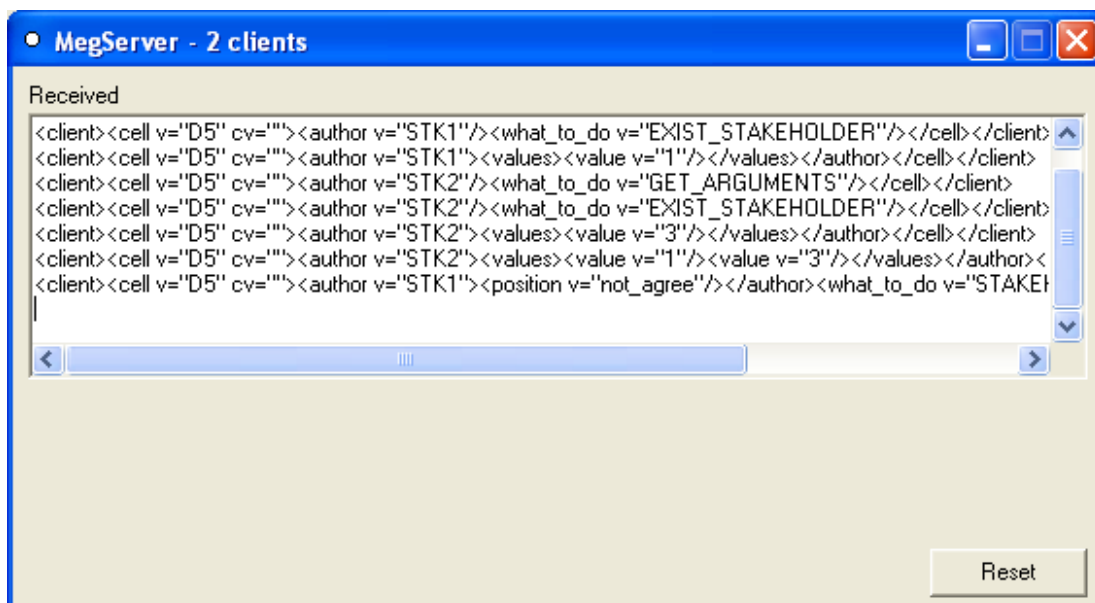


Figura 6.13 – Formulário – MEGSERVER

Segue-se o formato das mensagens XML entre a ferramenta MEGCLIENT e o servidor MEGSERVER.

Função 1 Obter argumentos do repositório de dados (MEGARG)

A função 1 é chamada quando se invoca o MEGCLIENT para uma célula pela primeira vez. Os argumentos ficam guardados em memória enquanto o MEGCLIENT da célula estiver activo. Quando o MEGCLIENT é fechado, o formulário é colocado em não visível, e quando invocado é colocado a visível, pelo que não é necessário obter os argumentos novamente. O cliente constrói uma mensagem XML em que refere a célula

(CELL), o stakeholder (AUTHOR) e o que é que o servidor deve fazer com a mensagem – neste caso é obter argumentos (GET_ARGUMENTS).

Pedido de MEGCLIENT: Obter argumentos

```
<client>
  <cell v = CELL cv = "">
    <author v = AUTHOR />
    <what_to_do v = "GET_ARGUMENTS" />
  </cell>
</client>
```

O servidor MEGSERVER obtém os argumentos do repositório de dados (tabela MEGARG) e constrói a mensagem XML em que refere a célula (CELL), o stakeholder (AUTHOR) e os itens (CATEGORY) e sub-itens (SUB-CATEGORY) mostrados na caixa Arguments do MEGCLIENT.

Resposta de MEGSERVER: Fornecer argumentos

```
<server>
  <cell v = CELL cv = "">
    <author v = AUTHOR />
    <arguments>
      <category v = "My own arguments">
        <sub_category v = SUB_CATEGORY d = ""/>
        ..
      </category>
      <category v = "Risk based arguments">
        <sub_category v = SUB_CATEGORY d = ""/>
        ..
      </category>
```

```
</arguments>
  <what_to_do v = "GET_ARGUMENTS" />
</cell>
</server>
```

Função 2 Obter máscara para formulário

O formulário do MEGCLIENT antes de se tornar visível, verifica se existe em célula a identificação do stakeholder (AUTHOR) que o invocou. Se não existe, o formulário apresenta a *fase de pré-negociação*. Caso contrário, apresenta a *fase de negociação*, ou de *pós-negociação*.

Pedido de MEGCLIENT: Existe stakeholder

```
<client>
  <cell v = CELL cv = "">
    <author v = AUTHOR />
    <what_to_do v = "EXIST_STAKEHOLDER" />
  </cell>
</client>
```

A máscara para o formulário será conforme a indicação da tag <stage>:

Resposta de MEGSERVER: Existe stakeholder

```
<server>
  <cell v = CELL cv = "">
    <author v = AUTHOR>
      <stage v = { PRE_NEGOTIATION | NEGOTIATION |
        POST-SETTLEMENT } />
    </author>
  </cell>
```

```
</server>
```

Função 3 Inserção de um valor em célula com CÉLULA em fase de pré-negociação

O formulário apresenta a fase de *pré-negociação* em que é possível inserir um e um só valor (um único valor para a tag <values>).

Pedido de MEGCLIENT: Inserir um valor

```
<client>
  <cell v = CELL cv = "">
    <author v = AUTHOR >
      <values>
        <value v = VALUE />
      </values>
    </author>
  </cell>
</client>
```

Quando o MEGCLIENT recebe a mensagem do servidor MEGSERVER, o formulário apresenta a *fase de negociação* conforme indicação da tag <stage>. A caixa Current Situation apresenta a *questão* com o valor da tag <issue> .

Resposta de MEGSERVER: Valor inserido

```
<server>
  <cell v = CELL cv = "">
    <author v = AUTHOR>
      <stage v = "NEGOTIATION" />
      <values>
        <value v = VALUE />
      </values>
```

```
</author>
<current_situation>
  <issue v = VALUE />
</current_situation>
</cell>
</server>
```

Função 4 Inserção de um valor em célula com CÉLULA em fase de negociação

O formulário apresenta a fase de *negociação* em que é possível inserir mais do que um valor (vários valores para a tag <values>). A presença da tag <i_will_agree> significa que a opção I'll agree foi seleccionada.

Pedido de MEGCLIENT: Valor inserido

```
<client>
  <cell v = CELL cv = "">
    <author v = AUTHOR >
      <values>
        <value v = VALUE />
        ..
      </values>
      <i_will_agree/>
    </author>
  </cell>
</client>
```

A função 4 trata a situação em que a inserção de novos valores não criou nenhuma solução que passaria à *fase de pós-negociação*.

O servidor MEGSERVER envia mensagens diferentes para o MEGCLIENT que o invocou e para os restantes que estão activos para essa célula. Quem fez o pedido recebe uma mensagem que contém informação dos valores e opção I'll agree seleccionados e situação corrente; os restantes, recebem informação referente apenas à situação corrente para a actualizar (*tag* <what_to_do>).

Resposta de MEGSERVER: Valor inserido

a) para o MEGCLIENT que fez pedido de inserção de valor

```
<server>
  <cell v = CELL cv = "">
    <author v = AUTHOR>
      <stage v = "NEGOTIATION" />
      <values>
        <value v = VALUE />
      </values>
    </author>
    <i_will_agree/>
    <current_situation>
      <issue v = VALUE >
        <positions>
          <position v= { IN_FAVOR | AGAINST } >
            <author v = AUTHOR />
          </position>
          ..
        </positions>
      </issue>
    </current_situation>
  </cell>
</server>
```

b) para os outros MEGCLIENT que estão activos para a mesma célula

```
<server>
  <cell v = CELL cv = "">
    <what_to_do v = "UPDATE_CURRENT_SITUATION" />
    <stage v = "NEGOTIATION" />
    <current_situation>
      <issue v = VALUE >
        <positions>
          <position v = { IN_FAVOR | AGAINST } >
            <author v = AUTHOR />
          </position>
          ..
        </positions>
      </issue>
    </current_situation>
  </cell>
</server>
```

Função 5 Inserção de um valor em célula com CÉLULA em fase de negociação e que passa a fase de pós-negociação

Semelhante à função 4 para a situação descrita de a) e b), mas a mensagem do servidor MEGSERVER indica que se deve passar à *fase de pós-negociação* conforme indicação da tag <stage>.

No entanto, a função 5 tem de considerar a situação de a opção I'll agree estar seleccionada. Neste caso, não passa ao formulário da *fase de pós-negociação* para obter a opção CONCORDO / NÃO CONCORDO.

Pedido de MEGCLIENT: Valor inserido

```
<client>
  <cell v = CELL cv = "">
    <author v = AUTHOR >
      <values>
        <value v = VALUE />
      </values>
      <i_will_agree/>
    </author>
  </cell>
</client>
```

Resposta de MEGSERVER: Valor inserido**a) para o MEGCLIENT que fez pedido de inserção de valor**

```
<server>
  <cell v = CELL cv = "">
    <author v = AUTHOR >
      <stage v = "POST_SETTLEMENT" />
      <values>
        <value v = VALUE />
        ..
      </values>
    </author>
    <current_situation>
      <issue v = VALUE >
        <positions>
          <position v = { IN_FAVOR | AGAINST } >
            <author v = AUTHOR />
```

```

    </position>
    ..
  </positions>
</issue>
<issue_proposed v = VALUE >
  <positions>
    <position v = { IN_FAVOR | AGAINST } >
      <author v = AUTHOR />
    </position>
    ..
  </positions>
</issue_proposed>
</current_situation>
<solution_proposed>
  <solution v = { ISSUE | ISSUE_PROPOSED} />
  <authors/>
</solution_proposed>
</cell>
</server>

```

b) para os outros MEGCLIENT que estão activos para a mesma célula

```

<server>
  <cell v = CELL cv = "">
    <what_to_do v = "UPDATE_CURRENT_SITUATION" />
    <stage v = "POST_SETTLEMENT" />
    <current_situation>
      <issue v = VALUE>
        <positions>
          <position v = { IN_FAVOR | AGAINST } >
            <author v = AUTHOR />

```

```

    </position>
    ..
  </positions>
</issue>
<issue_proposed v = VALUE >
  <positions>
    <position v = { IN_FAVOR | AGAINST } >
      <author v = AUTHOR />
    </position>
    ..
  </positions>
</issue_proposed>
</current_situation>
<solution_proposed>
  <solution v = { ISSUE | ISSUE_PROPOSED} />
  <authors/>
</solution_proposed>
</cell>
</server>

```

c) para os MEGCLIENT com opção I'll agree seleccionada

```

<server>
  <cell v = CELL cv = "">
    <what_to_do v = "UPDATE_JUST_CURRENT_SITUATION" />
    <stage v = "POST_SETTLEMENT" />
    <current_situation>
      <issue v = VALUE >
        <positions>
          <position v = { IN_FAVOR | AGAINST } >
            <author v = AUTHOR />

```

```
</position>
..
</positions>
</issue>
<issue_proposed v = VALUE >
  <positions>
    <position v = { IN_FAVOR | AGAINST } >
      <author v = AUTHOR />
    </position>
  ..
</positions>
</issue_proposed>
</current_situation>
<solution_proposed>
  <solution v = { ISSUE | ISSUE_PROPOSED} />
  <authors/>
</solution_proposed>
<situation_not_agree>
  <solution_issue v = { ISSUE | ISSUE_PROPOSED |
ISSUE_STAKEHOLDER_FIRM } >
  <solution_value v = VALUE />
  <solution_author v = AUTHOR />
</solution_issue>
</situation_not_agree>
</cell>
</server>
```

Função 6 Stakeholder CONCORDA em célula com CÉLULA em fase de pós-negociação

O stakeholder seleccionou a opção CONCORDO. Se todos os stakeholders têm a mesma opção, o servidor MEGSERVER devolve uma mensagem ao MEGCLIENT que avisa cada stakeholder.

Pedido de MEGCLIENT: Opção CONCORDO seleccionada

```
<client>
  <cell v = CELL cv = "">
    <author v = AUTHOR >
      <position v = "agree" />
    </author>
    <what_to_do v = "STAKEHOLDER_AGREE" />
  </cell>
</client>
```

Resposta de MEGSERVER: Quando todos os stakeholders de célula tem opção CONCORDO

```
<server>
  <cell v = CELL cv = "">
    <author v = AUTHOR />
    <what_to_do v = "STAKEHOLDER_ALL_AGREE" />
    <current_situation>
      <issue v = VALUE />
    </current_situation>
  </cell>
</server>
```

Função 7 Stakeholder NÃO CONCORDA em célula com CÉLULA em fase de pós-negociação

Quando existe um stakeholder que selecciona a opção NÃO CONCORDO, o servidor MEG envia a todos os MEGCLIENT afectados à célula, a indicação de que devem voltar à *fase de negociação*. Dado que o formulário mostra a *fase de pós-negociação* através da colocação das caixas Value e Arguments a invisíveis, a passagem à *fase de negociação* consiste em colocar essas caixas novamente visíveis (como as opções das caixas não foram alteradas, o estado anterior mantém-se). A caixa Current Situation é actualizada.

Pedido de MEGCLIENT: Opção NÃO CONCORDO seleccionada

```
<client>
  <cell v = CELL cv = "">
    <author v = AUTHOR >
      <position v = "not_agree" />
    </author>
    <what_to_do v = "STAKEHOLDER_NOT_AGREE" />
  </cell>
</client>
```

A situação a que não se chegou a consenso é guardada na *tag* <situation_not_agree> (contém origem da solução proposta, valor e autor).

Resposta de MEGSERVER: Quando um stakeholder de célula tem opção NÃO CONCORDO

```
<server>
  <cell v = CELL cv = "">
    <what_to_do v = "STAKEHOLDER_NOT_AGREE" />
    <stage v = "NEGOTIATION" />
```

```

<current_situation>
  <issue v = VALUE >
    <positions>
      <position v = { IN_FAVOR | AGAINST } >
        <author v = AUTHOR />
      </position>
      ..
    </positions>
  </issue>
  <issue_proposed v = VALUE >
    <positions>
      <position v = { IN_FAVOR | AGAINST } >
        <author v = AUTHOR />
      </position>
      ..
    </positions>
  </issue_proposed>
</current_situation>
<situation_not_agree>
  <solution_issue v = { ISSUE | ISSUE_PROPOSED |
ISSUE_STAKEHOLDER_FIRM } >
    <solution_value v = VALUE />
    <solution_author v = AUTHOR />
  </solution_issue>
</situation_not_agree>
</cell>
</server>

```

Função 8 Stakeholder tenta propor um valor para célula com CÉLULA em fase de negociação

Apenas é possível inserir um e um só valor (um único valor para a tag <values>) como valor proposto por um stakeholder. Se estiver seleccionada a opção I'll agree o MEGCLIENT não passa à *fase de pós-negociação*.

Pedido de MEGCLIENT: Opção FIRM seleccionada

```
<client>
  <cell v = CELL cv = "">
    <author v = AUTHOR >
      <values>
        <value v = VALUE />
      </values>
      <i_will_agree/>
    </author>
    <what_to_do v = "STAKEHOLDER_FIRM" />
  </cell>
</client>
```

Na *fase de pós-negociação*, a caixa Current Situation apresenta o numero de vezes que o stakeholder seleccionou a opção FIRM e / ou LOCK, no total de vezes em que essas opções foram usadas. Esta informação é obtida a partir do repositório de dados (tabela ALOCTAB).

Resposta de MEGSERVER: Valor proposto por stakeholder

a) para o MEGCLIENT que fez pedido de inserção de valor

```
<server>
  <cell v = CELL cv = "">
    <author v = AUTHOR >
      <stage v = "POST_SETTLEMENT" />
```



```
</author>
<current_situation>
  <issue v = VALUE >
    <positions>
      <position v = { IN_FAVOR | AGAINST } >
        <author v = AUTHOR />
      </position>
      ..
    </positions>
  </issue>
  <issue_proposed v = VALUE >
    <positions>
      <position v = { IN_FAVOR | AGAINST } >
        <author v = AUTHOR />
      </position>
      ..
    </positions>
  </issue_proposed>
  <issue_stakeholder_firm v = VALUE>
    <author v = AUTHOR />
  </issue_stakeholder_firm>
</current_situation>
<solution_proposed>
  <solution v = "ISSUE_STAKEHOLDER_FIRM" />
  <authors/>
</solution_proposed>
<a_t_aloctx>
  <author v = AUTHOR >
  <firm v = NTIMES />
  <lock v = NTIMES />
```

```
</author>
<total>
  <firm v = NTIMES />
  <lock v = NTIMES />
</total>
</a_t_aloctxab>
</cell>
</server>
```

b) para os outros MEGCLIENT que estão activos para a mesma célula

```
<server>
<cell v = CELL cv = "">
  <what_to_do v = "UPDATE_CURRENT_SITUATION" />
  <stage v = "POST_SETTLEMENT" />
  <current_situation>
    <issue v = VALUE>
      <positions>
        <position v = { IN_FAVOR | AGAINST } >
          <author v = AUTHOR />
        </position>
        ..
      </positions>
    </issue>
    <issue_proposed v = VALUE>
      <positions>
        <position v = { IN_FAVOR | AGAINST } >
          <author v = AUTHOR />
        </position>
        ..
      </positions>
```

```

</issue_proposed>
<issue_stakeholder_firm v = VALUE>
  <author v = AUTHOR />
</issue_stakeholder_firm>
</current_situation>
<solution_proposed>
  <solution v = "ISSUE_STAKEHOLDER_FIRM" />
  <authors/>
</solution_proposed>
<a_t_aloactab>
  <author v = AUTHOR >
    <firm v = NTIMES />
    <lock v = NTIMES />
  </author>
  <total>
    <firm v = NTIMES />
    <lock v = NTIMES />
  </total>
</a_t_aloactab>
</cell>
</server>

```

c) para os MEGCLIENT com opção I'll agree seleccionada

```

<server>
  <cell v = CELL cv = "">
    <what_to_do v = "UPDATE_JUST_CURRENT_SITUATION" />
    <stage v = "POST_SETTLEMENT" />
    <current_situation>
      <issue v = VALUE >
      <positions>

```

```

    <position v = { IN_FAVOR | AGAINST } >
      <author v = AUTHOR />
    </position>
    ..
  </positions>
</issue>
<issue_stakeholder_firm v = VALUE>
  <author v = AUTHOR />
</issue_stakeholder_firm>
</current_situation>
<solution_proposed>
  <solution v = "ISSUE_STAKEHOLDER_FIRM" />
  <authors/>
</solution_proposed>
<situation_not_agree>
<solution_issue v = { ISSUE | ISSUE_PROPOSED |
ISSUE_STAKEHOLDER_FIRM } >
  <solution_value v = VALUE />
  <solution_author v = AUTHOR />
</solution_issue>
</situation_not_agree>
<a_t_aloctab>
  <author v = AUTHOR >
    <firm v = NTIMES />
    <lock v = NTIMES />
  </author>
<total>
  <firm v = NTIMES />
  <lock v = NTIMES />
</total>

```

```
</a_t_aloetab>  
</cell>  
</server>
```

Função 9 Stakeholder impossibilita a CÉLULA em fase de negociação passar para a fase de pós-negociação

A opção LOCK origina uma mensagem de aviso de que não é possível passar à fase de *pós-negociação* com a identificação do stakeholder responsável por essa situação (*tag* <stakeholder_lock>).

Pedido de MEGCLIENT: Opção LOCK seleccionada

```
<client>  
  <cell v = CELL cv = "">  
    <author v = AUTHOR />  
    <what_to_do v = "STAKEHOLDER_LOCK" />  
  </cell>  
</client>
```

Resposta de MEGSERVER: Opção LOCK seleccionada com identificação do stakeholder

```
<server>  
  <cell v = CELL cv = "">  
    <author v = AUTHOR />  
    <what_to_do v = "STAKEHOLDER_LOCK" />  
    <current_situation>  
      <issue v = VALUE >  
        <positions>  
          <position v = { IN_FAVOR | AGAINST } >
```

```

    <author v = AUTHOR />
  </position>
  ..
</positions>
</issue>
<issue_proposed v = VALUE >
  <positions>
    <position v = { IN_FAVOR | AGAINST } >
      <author v = AUTHOR />
    </position>
    ..
  </positions>
</issue_proposed>
</current_situation>
<stakeholder_lock v = AUTHOR />
<situation_not_agree>
  <solution_issue v = { ISSUE | ISSUE_PROPOSED |
ISSUE_STAKEHOLDER_FIRM } >
    <solution_value v = VALUE />
    <solution_author v = AUTHOR />
  </solution_issue>
</situation_not_agree>
</cell>
</server>

```

Função 10 Stakeholder restabelece possibilidade de passagem da CÉLULA em fase de negociação para a fase de pós-negociação

Quando o stakeholder responsável pela opção LOCK retira a opção, o servidor MEGSERVER envia uma mensagem a todos os MEGCLIENT com essa informação.

Se a opção LOCK inibia uma situação que possibilitava a passagem à *fase de pós-negociação*, ao ser retirada essa opção pode acontecer que automaticamente se passe à *fase de pós-negociação* (conforme tag <stage>).

Pedido de MEGCLIENT: Retira opção LOCK seleccionada

```
<client>
  <cell v = CELL cv = "">
    <author v = AUTHOR />
    <what_to_do v= "STAKEHOLDER_UNLOCK" />
  </cell>
</client>
```

Resposta de MEGSERVER: Opção LOCK retirada

```
<server>
  <cell v = CELL cv = "">
    <author v = AUTHOR />
    <what_to_do v = "STAKEHOLDER_UNLOCK" />
    <stage v = {NEGOTIATION | POST-SETTLEMENT} />
    <current_situation>
      <issue v = VALUE >
        <positions>
          <position v = { IN_FAVOR | AGAINST } >
            <author v = AUTHOR />
          </position>
          ..
        </positions>
      </issue>
      <issue_proposed v = VALUE >
        <positions>
```

```
<position v = { IN_FAVOR | AGAINST } >
  <author v = AUTHOR />
</position>
..
</positions>
</issue_proposed>
</current_situation>
<situation_not_agree>
  <solution_issue v = { ISSUE | ISSUE_PROPOSED |
ISSUE_STAKEHOLDER_FIRM } >
    <solution_value v = VALUE />
    <solution_author v = AUTHOR />
  </solution_issue>
</situation_not_agree>
</cell>
</server>
```

O servidor MEGSERVER foi elaborado em VISUALBASIC 6.0 Enterprise Edition com SP5 e tem como base a parte *servidor* de [Planetsourcecode, 2003]. A gestão do acesso ao repositório de dados é feito por intermédio de ADO (ActiveX Data Objects) [Microsoft, 2004].

6.2.4 RTDSERVER

O servidor RTDSERVER é responsável pela sincronização dos dados na matriz Casa da Qualidade para Software em cada folha de cálculo. Consiste num servidor de automação COM (Component Object Model) que implementa um interface IRtdServer [Cornell, 2001].

O EXCEL comunica com o servidor RTDSERVER para obter os dados presentes no repositório. É necessário um identificador único (tópico) que identifique um dado que resida no repositório – no sistema MEG, o tópico consiste na coordenada de uma célula.

O repositório é monitorizado pelo servidor RTDSERVER a cada t segundos (no sistema MEG, $t = 5$ segundos), onde se observa na tabela MEGREP o atributo cy de cada tag XML que corresponde à identificação de uma célula. Os valores possíveis para cy são um valor de $\{ 0, 1, 3, 9 \}$ ou $\{ ?, F, L \}$, valor esse que o servidor RTDSERVER retorna à CÉLULA de EXCEL que lhe deu por tópico a coordenada de CÉLULA.

O servidor RTDSERVER foi construído em VISUALBASIC 6.0 Enterprise Edition com SP5. É condição necessária ter instalado o MSXML para manipulação do XML e configurar o DCOM (Distributed Component Object Model) [Microsoft, 2000] [Microsoft, 2003] para a comunicação entre o EXCEL e o servidor RTDSERVER remoto [Microsoft, 2001]. A gestão do acesso ao repositório de dados é feito por intermédio de ADO (ActiveX Data Objects) [Microsoft, 2004].

6.2.5 REPOSITÓRIO DE DADOS

O repositório de dados (MEGREPOSITORY.MDB) é uma base de dados Microsoft ACCESS composta por 3 tabelas que contém registos em formato XML: a) MEGREP, é a tabela principal em que são guardados os dados referentes aos valores e situação actual de cada CÉLULA por stakeholder; b) MEGARG, guarda os argumentos baseados numa ontologia; c) ALOCTAB, com informações referentes às opções de menu FIRM e LOCK.

O formato das tabelas do repositório consiste em duas colunas que guardam respectivamente a data e hora da inserção do registo (tipo de dados: Data/hora) e o XML (tipo de dados: Memo). O formato do XML de cada tabela é descrito de seguida.

MEGREP

Na tabela MEGREP são guardadas indicações do conjunto de células que contém valores e respectiva situação em que se encontra cada célula:

```
<cells>
  <cell> .. </cell>
  ..
</cells>
```

Para uma CÉLULA, temos as seguintes secções:

```
<cell v = CELL_COORDINATE cv = { 0 | 1 | 3 | 9 | ? | F | L }>
<author> .. </author>
<stage v = { NEGOTIATION | POST_SETTLEMENT } />
<current situation> .. </current situation>
<issue proposed> .. </issue proposed>
<solution proposed> .. </solution proposed>
<situation not agree> .. </situation not agree>
</cell>
```

Podendo haver vários stakeholders para uma célula, a secção *author* contém os valores (value) inseridos e a tag *<i will agree>* se essa opção tiver sido escolhida na fase de negociação:

```
<author>
  <values>
    <value v = VALUE />
    ..
  </values>
  <i will agree />
</author>
```

A secção *stage* apenas faz referencia às fases de *negociação* (negotiation) e de *pós-negociação* (post-settlement) em que uma CÉLULA se encontra. Assume-se que se não existem valores inseridos é porque a CÉLULA se encontra na *fase de pré-negociação*.

A secção *current situation* é composta pelas *questões* (issue) e *posições* (position) dos stakeholders relativamente a esse valor:

```
<current situation>
  <issue v = VALUE >
    <positions>
      <position v = { IN_FAVOR | AGAINST } />
      ..
    </positions>
  </issue>
  <issue proposed v = VALUE>
    <positions>
      <position v = { IN_FAVOR | AGAINST } />
      ..
    </positions>
  </issue proposed>
</current situation>
```

A secção *solution proposed* guarda a solução que será sujeita a consenso, quer tenha sido obtida pela opção de menu FIRM, quer por uma das *questões* conter todas as posições a favor de um valor:

```
<solution proposed>
  <solution v = { ISSUE | ISSUE_PROPOSED
| ISSUE_STAKEHOLDER_FIRM } />
  <authors>
    <author v = AUTHOR>
```

```

        <position v = { AGREE | NOT_AGREE } />
    </author>
    ..
</authors>
</solution proposed>

```

A secção *situation not agree* guarda a situação em que não foi possível chegar a consenso (valor em causa e modo de obtenção desse valor):

```

<situation not agree>
    <solution issue v = { ISSUE | ISSUE_PROPOSED
    | ISSUE_STAKEHOLDER_FIRM }
    <solution value v = VALUE />
    </solution issue>
</situation not agree>

```

Quando é guardada a tag <situation not agree>, apaga-se a tag <solution proposed>.

MEGARG

A tabela MEGARG guarda um conjunto de palavras reservadas que servem de base aos *argumentos* possíveis dos stakeholders.

```

<arguments>
    <category v="My own arguments">
        <sub_category v = SUB_CATEGORY
        d = SUB_CATEGORY_DESC/>
        ...
    </category>
    <category v = "Risk based arguments">
        <sub_category v = SUB_CATEGORY
        d = SUB_CATEGORY_DESC/>

```

```
    ...
  </category>
</arguments>
```

No caso da figura 6.4, para a categoria *Risk based arguments*, tem-se que `SUB_CATEGORY = { Maintainability | Reliability | Safety | Security | Human Factors | Specifications }`.

Este conjunto `SUB_CATEGORY` é apresentado apenas a título ilustrativo como possibilidade de inserção de argumentos. A obtenção de um conjunto válido exigiria uma análise cuidada, que está fora do âmbito deste trabalho.

ALOCTAB

A tabela `ALOCTAB` armazena a informação referente ao número de vezes que foram utilizadas as opções de menu `FIRM` e `LOCK`, quer no total quer por stakeholder:

```
<aloc>
  <total>
    <firm v = NTIMES />
    <lock v = NTIMES />
  </total>
  <author v = AUTHOR>
    <firm v = NTIMES />
    <lock v = NTIMES />
  </author>
  ..
</aloc>
```

Capítulo 7

Avaliação preliminar

7.1. Contexto

Foram realizadas experiências onde se pretende avaliar a negociação de requisitos no sistema MEG. São descritas duas das experiências efectuadas:

- experiência 1, onde participaram 3 stakeholders, decorreu na Unidade de Informática do Centro Nacional de Pensões (CNP);
- experiência 2, em que participaram 2 stakeholders, ocorreu na residência de um dos stakeholders.

Os casos apresentados restringem-se apenas à Casa da Qualidade para Software (software-HoQ).

7.2. Experiência 1

Stakeholders

O stakeholder A detém mais de 30 anos de experiência no domínio da informática e prática na negociação de requisitos de software com *outsourcing*. O stakeholder B detém 6 anos de experiência em análise de sistemas. O stakeholder Z desempenha o papel de um dos responsáveis pelo desenvolvimento do sistema (João Ramires), actuando simultaneamente como observador.

Tarefas

O sistema pretendido consiste numa integração de um novo tipo de cálculo na aplicação de Cálculo de Prestações. Como auxílio na identificação dos itens da Voz do Engenheiro, recorreu-se ao padrão ISO/IEC 9126 [ISO/IEC9126, 1991].

O objectivo do padrão ISO/IEC 9126 é providenciar uma estrutura de avaliação da qualidade do software. Este padrão não define requisitos para software mas sim um modelo de qualidade que pode ser aplicável a todo o tipo de software, sendo dividido em seis características (Funcionalidade, Confiabilidade, Usabilidade, Eficiência, Manutenção e Portabilidade) que por sua vez se desdobram em sub-características.

O padrão ISO/IEC 9126 pode ser usado para potenciar a qualidade da apresentação de soluções por parte dos stakeholders responsáveis pelo desenvolvimento do sistema. O processo de Categorização da Voz do Engenheiro referido no capítulo 3, pode portanto ser auxiliado pelo ISO/IEC 9126.

Procedimento

Realizou-se uma entrevista com o stakeholder que pretendia o sistema (stakeholder A) por um dos stakeholders responsáveis pelo desenvolvimento do sistema (stakeholder Z). A partir desta entrevista, foram obtidas a Voz do Cliente e a Voz do Engenheiro para a construção da Casa da Qualidade para Software, agrupadas respectivamente pelas características e sub-características do padrão ISO/IEC 9126 pelo stakeholder Z.

Obtida a matriz SQFD, foi realizada uma sessão de formação aos stakeholders A e B com duração de uma hora antes da utilização do sistema MEG. Esta sessão teve como objectivo a familiarização dos conceitos utilizados, nomeadamente:

- uma breve explicação do método SQFD;
- as atitudes que os stakeholders podem tomar num processo de negociação;
- a forma como o sistema mostra as posições dos stakeholders sem expor as opções individuais;
- a forma de justificar as posições assumidas;
- os valores de correlação serem obtidos por consenso;
- o sistema poder propor um valor de correlação atendendo ao contexto.

Ambiente físico

A experiência 1 realizou-se com recurso a dois computadores com as aplicações cliente em salas separadas. As negociações foram feitas entre pares alternados de stakeholders, não havendo contacto extra sistema MEG durante a experiência entre cada par.

7.2.1 Dados recolhidos

São apresentados excertos da entrevista com o stakeholder A para a obtenção da Voz do Cliente:

- “Integração de um novo tipo de cálculo na aplicação Cálculo de Prestações”;
- “Fácil evolução e expansão; permita a adição de novas funcionalidades com um mínimo de alterações ao sistema existente”;
- “Manutenção futura; minimize as necessidades e peso da manutenção”;
- “Migração e contingência; planos detalhados de migração e de contingência”;
- “Passagem do conhecimento; documentação clara e completa de todo o sistema, para utilizadores finais e para os técnicos de TIC¹ (analistas, programadores, operadores)”;
- “Alterações nos processos existentes; definição clara das alterações que serão introduzidas nos processos existentes no cliente, quer utilizadores finais, quer técnicos de TIC do cliente”;
- “Formação dos utilizadores; garanta a formação adequada e atempada dos utilizadores, permitindo-lhes a utilização prévia do sistema em ambiente de treino e teste”.

¹ Tecnologias de Informação e Comunicação

Voz do Cliente

Os excertos da entrevista com o stakeholder A deram origem aos seguintes itens da Voz do Cliente agrupadas segundo *características* do padrão ISO/IEC 9126:

- Funcionalidade
 - “Novo tipo de cálculo C”²
 - “Integrado na aplicação Cálculo de Prestações”
- Confiabilidade
 - “Planos detalhados de contingência”
- Usabilidade
 - “Garanta a formação adequada e atempada dos utilizadores”
 - “Documentação clara e completa de todo o sistema”
 - “Definição clara das alterações que serão introduzidas nos processos existentes”
- Eficiência

Não aplicável à software-HoQ. Característica mensurável.
- Manutenção
 - “Permita a adição de novas funcionalidades com o mínimo de alterações”
 - “Minimize as necessidades e peso da manutenção”
- Portabilidade
 - “Planos detalhados de migração”

² Denominações de forma abstracta para exposição da experiência.

Voz do Engenheiro

Foram obtidos igualmente itens para a Voz do Engenheiro a partir dos excertos da entrevista com o stakeholder A, agrupadas segundo *sub-características* do padrão ISO/IEC 9126:

- Funcionalidade
 - Conveniência
 - Calcular as pensões P
 - Exactidão
 - Calcula e mostra os cálculos de pensão das entidades E1 e E2
 - Grava dados segundo perfil de utilizador
 - Interoperabilidade
 - Parte cliente da aplicação pode trabalhar em diferentes sistemas operativos
 - Concordância
 - Disponibiliza informação da legislação em que o cálculo C se baseia
 - Segurança
 - Exige autenticação dos utilizadores
- Confiabilidade
 - Maturidade
 - Completa um ciclo transaccional sem erros de execução
 - Tolerância a faltas
 - Continua a funcionar se os dados estiverem indisponíveis
 - Continua a funcionar se os dados estiverem mal gravados
 - Se servidor principal falha, responde servidor secundário

- Recuperação
 - Pode recuperar operações completadas antes de uma falha de corrente
- Usabilidade
 - Compreensão
 - Organização dos itens no ecrã
 - Esquemas e explicação do funcionamento do sistema online
 - Aprendizagem
 - Providencia dicas / ajuda
 - Providencia alerta das novas funcionalidades
 - Permite o uso do sistema teste com dados reais
 - Operabilidade
 - Opções disponíveis consoante privilégios de utilizador
- Eficiência

Não aplicável à software-HoQ. Sub-características mensuráveis.
- Manutenção
 - Análise
 - Apresentação de como se efectuam os cálculos
 - Mutabilidade
 - Permitir a alteração de códigos / coeficientes sem alteração do programa
 - Esquema modular de programação
 - Reutilização de módulos
 - Estabilidade
 - Dados facilmente acessíveis em qualquer definição de ecrã
 - Impressão de dados igual em qualquer tipo de impressora
- Portabilidade

- Adaptabilidade
 - Dados exportáveis para Excel
- Instalação
 - Norma interna “Passagem de Programas a Produção” / 2004
- Conformidade
 - Norma interna 002/2004
- Substituição

Não aplicável neste caso.

Obtida a Voz do Cliente e a Voz do Engenheiro, foi construída a Casa de Qualidade para software.

Casa da Qualidade para Software

The screenshot shows an Excel spreadsheet titled 'QFD Software-HoQ - ISO/IEC 9126'. The main table is a matrix with rows representing customer requirements and columns representing technical characteristics. The matrix is organized into sections: 'Funcionalidade', 'Confiabilidade', 'Usabilidade', 'Manutenção', and 'Portabilidade'. Each cell contains a numerical rating from 1 to 9. Below the main table, there are detailed descriptions for each technical characteristic.

		Funcionalidade	Confiabilidade	Usabilidade	Manutenção	Portabilidade
5		Conveniência Exactidão	Interoperabilidade Concordância Segurança	Maturidade Tolerância a falhas	Recuperação Compreensão Aprendizagem	Operabilidade Análise Mutabilidade Estabilidade Adaptabilidade Instalação Conformidade
6		Calcular as pensões. P	Calcula e mostra os cálculos da pensão das entidades E1 e E2 Grava dados segundo perfil de utilizador	Parte cliente da aplicação pode trabalhar em diferentes sistemas operativos Disponibiliza informação da legislação em que o cálculo C se baseia	Exige autenticação dos utilizadores Completa um ciclo transaccional sem erros de execução	Continua a funcionar se os dados estiverem indisponíveis Continua a funcionar se os dados estiverem mal gravados Se servidor principal falha, responde servidor secundário Pode recuperar operações completadas antes de uma falha de corrente Organização dos itens no ecrã Esquemas e explicação do funcionamento do sistema online Providencia dicas / ajuda Providencia alerta das novas funcionalidades Permite o uso do sistema teste com dados reais Opções disponíveis consoante privilégios de utilizador Apresentação de como se efectuam os cálculos Permitir a alteração de códigos / coeficientes sem alteração do programa Esquema modular de programação Realização de módulos Dados facilmente acessíveis em qualquer definição de ecrã Impressão de dados igual em qualquer tipo de impressora Dados exportáveis para Excel Norma interna 'Passagem de Programas a produção' /2004 Norma interna 002 /2004
7	Funcionalidade	Novo tipo de cálculo C	9	3		
8	Confiabilidade	Planos detalhados de contingência				
9	Usabilidade	Formação adequada e atempada dos utilizadores Documentação clara e completa de todo o sistema Definição clara das alterações nos processos existentes				
10	Manutenção	Novas funcionalidades com o mínimo de alterações Minimize as necessidades e peso da manutenção				
11	Portabilidade	Planos detalhados de migração				

Figura 7.1 - Casa da Qualidade para Software (projecto Cálculo de Prestações) representada no sistema MEG

A representação da Casa de Qualidade para Software é apresentada na figura 7.1. reflectindo o resultado das negociações entre stakeholders. A figura 7.2. mostra o formulário na fase de negociação para o valor da correlação entre “Planos detalhados

de contingência” (Confiabilidade) e “Pode recuperar operações completadas antes de uma falha de corrente” (Confiabilidade – Recuperação).

A ontologia que serve de suporte aos argumentos é composta por:

- Soluções: Funcionalidade, Dificuldade, Interfaces internos, Performance, Testável, Constrangimentos de hardware e Software não desenvolvido;
- Recursos: Calendarização, Pessoas, Orçamento e Facilidades.

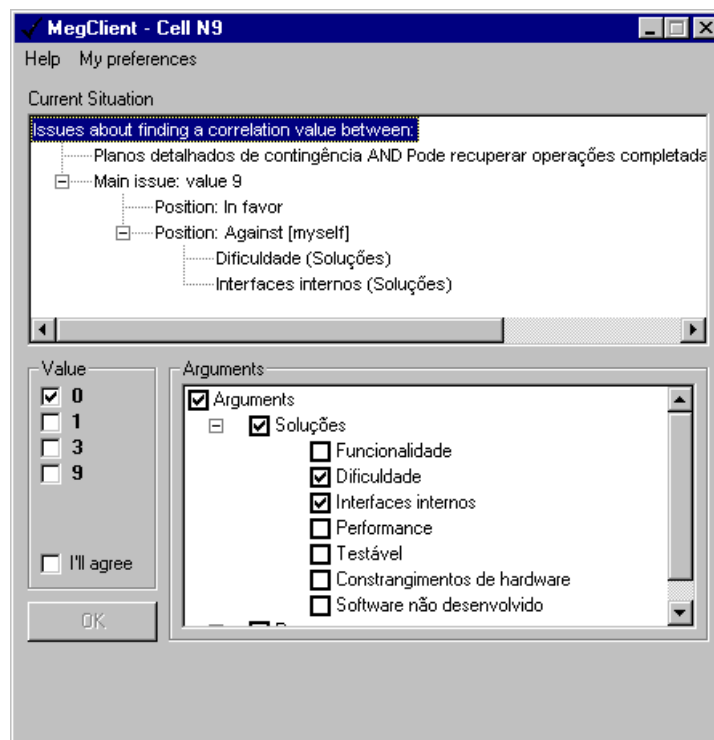


Figura 7.2 – Formulário – fase de negociação

A tabela 7.1. descreve o hardware, software e configuração do DCOM presente em cada computador utilizado na experiência.

Tabela 7.1 – Recursos utilizados na experiência no CNP

HARDWARE	SOFTWARE	
	Programas	Configuração DCOM para RTDSERVER
SERVIDOR Pentium II 350Mhz 64MB	Windows NT MEGSERVER RTDSERVER EXCEL 2002 ACCESS 97	Propriedades predefinidas: <u>Activar DCOM neste computador</u> Nível de autenticação predefinido: <u>Nenhum</u> Nível de representação predefinido: <u>Representar</u> Nível de autenticação: <u>Nenhum</u> <u>Executar a aplicação neste computador</u> Iniciar permissões, Permissões de acesso, Permissões de configuração: <u>INTERACTIVE, SYSTEM, Todos</u> Identidade: <u>O utilizador interactivo</u>
CLIENTE 1 Pentium IV 2.6Ghz 512MB	Windows XP Pro MEGCLIENT RTDSERVER EXCEL 2002	Propriedades predefinidas: <u>Activar DCOM neste computador</u> Nível de autenticação predefinido: <u>Nenhum</u> Nível de representação predefinido: <u>Representar</u> Nível de autenticação: <u>Nenhum</u>
CLIENTE 2 Pentium II 350Mhz 64MB	Windows NT MEGCLIENT RTDSERVER EXCEL 2002	<u>Executar a aplicação no seguinte computador CNPI307</u> Iniciar permissões, Permissões de acesso, Permissões de configuração: <u>INTERACTIVE, SYSTEM, Todos</u> Identidade: <u>O utilizador que inicia</u>

7.3. Experiência 2

Stakeholders

O stakeholder C exerce as funções de coordenador da área de sistemas de informação numa empresa de telecomunicações. O stakeholder D é analista / programador – área

de estatística e investigação operacional. O stakeholder Z (João Ramires) actua neste caso apenas como observador.

Tarefas

Nesta experiência foi usada a Casa de Qualidade apresentada na figura 6.1. em que se pretende avaliar a aplicação junto a dois stakeholders, sem recorrer a um caso real como o elaborado na experiência 1.

Procedimento

Foi apresentada a Casa da Qualidade da figura 6.1. seguindo-se uma sessão de formação aos stakeholders C e D com duração de um quarto de hora antes da utilização do sistema, optando-se por apresentar os conceitos à medida que eram necessários e num nível de detalhe inferior ao apresentado na experiência 1.

Ambiente físico

A experiência 2 foi realizada num único computador. O stakeholder C usou o sistema normalmente, enquanto o stakeholder D acedia aos valores das células via *versão teste* do MEGCLIENT (tal como mostrado no exemplo da figura 6.5). O computador foi usado alternadamente, sendo a mesma folha de EXCEL utilizado para visualizar as alterações obtidas, não havendo comunicação entre os dois stakeholders durante a experiência.

7.3.1 Dados recolhidos

Não foram recolhidos dados já que a experiência se baseou na Voz do Cliente e na Voz do Engenheiro apresentada na figura 6.1.

A ontologia utilizada que serviu como ilustração ao exemplo foi a apresentada na caixa Arguments da figura 6.3: My own arguments = { Stability, Completeness, Clarity, Validity, Feasibility, Precedent, Scale }; Risk based arguments = { Maintainability, Reliability, Safety, Security, Human Factors, Specifications }.

Dado que a aplicação foi executada localmente, não houve necessidade de configurar o DCOM. A máquina utilizada foi um Celeron a 2.2Ghz com 512Mb.

7.4. Análise de resultados

Segue-se a avaliação dos stakeholders A, B, C e D face à pergunta “O Sistema MEG suporta negociação de requisitos de software?”. As respostas aos formulários são apresentados de seguida.

Formulário de Stakeholder A

Assinalar com X na coluna correspondente

	☹	☺	☺	☺	☺
Funcionalidade	1	2	3	4	5
Conveniência Existem funções apropriadas para as tarefas específicas				X	
Exactidão Proporciona resultados / efeitos correctos				X	
Concordância De acordo com padrões (voto maioritário, consenso, etc.)				X	
Usabilidade					
Compreensão Esforço dos stakeholders para reconhecerem a lógica dos conceitos e sua aplicação			X		
Aprendizagem Esforço dos stakeholders para aprenderem a aplicação			X		
Operabilidade Esforço dos stakeholders para operarem e controlarem a operação			X		

Aspectos negativos
<ul style="list-style-type: none"> Muito lento Não é muito fácil de usar pelo utilizador comum
Aspectos positivos
<ul style="list-style-type: none"> Facilidade em encontrar um ponto de acordo Conhecer os argumentos dos outros para poder avaliar e eventualmente rever a minha posição

Formulário de Stakeholder B

Assinalar com X na coluna correspondente



Funcionalidade	1	2	3	4	5
Conveniência Existem funções apropriadas para as tarefas específicas				X	
Exactidão Proporciona resultados / efeitos correctos			X		
Concordância De acordo com padrões (voto maioritário, consenso, etc)				X	
Usabilidade					
Compreensão Esforço dos stakeholders para reconhecerem a lógica dos conceitos e sua aplicação				X	
Aprendizagem Esforço dos stakeholders para aprenderem a aplicação				X	
Operabilidade Esforço dos stakeholders para operarem e controlarem a operação			X		
Aspectos negativos					
<ul style="list-style-type: none"> Se dois stakeholders de três negociam um valor e chegam a acordo, esse valor fica visível na célula. Se outro par negocia outro valor para essa célula, um dos dois stakeholders iniciais não é avisado da alteração Sistema lento 					
Aspectos positivos					
<ul style="list-style-type: none"> Melhor compreensão do conjunto global de ideias dos stakeholders 					

Formulário de Stakeholder C

Assinalar com X na coluna correspondente



Funcionalidade	1	2	3	4	5
Conveniência Existem funções apropriadas para as tarefas específicas				X	
Exactidão Proporciona resultados / efeitos correctos			X		
Concordância De acordo com padrões (voto maioritário, consenso, etc.)				X	
Usabilidade					
Compreensão Esforço dos stakeholders para reconhecerem a lógica dos conceitos e sua aplicação			X		
Aprendizagem Esforço dos stakeholders para aprenderem a aplicação			X		
Operabilidade Esforço dos stakeholders para operarem e controlarem a operação				X	

Aspectos negativos
No que concerne ao modo como são exibidas as várias decisões de cada utilizador deveria existir um modo gráfico mais intuitivo, isto é, para o utilizador não é relevante saber o número de iterações do programa mas sim determinar num dado instante de tempo qual o estado de um dado requisito e quais os critérios que tiveram mais peso para esse estado
Aspectos positivos
O modelo de suporte à decisão, não obstante da forma como são exibidos os resultados, em termos práticos revela-se eficiente, no entanto para gestores de topo seria preferível implementar um interface mais gráfico

Formulário de Stakeholder D

Assinalar com X na coluna correspondente

	☹	☺	☺	☺	☺
Funcionalidade	1	2	3	4	5
Conveniência Existem funções apropriadas para as tarefas específicas				X	
Exactidão Proporciona resultados / efeitos correctos				X	
Concordância De acordo com padrões (voto maioritário, consenso, etc)					X
Usabilidade					
Compreensão Esforço dos stakeholders para reconhecerem a lógica dos conceitos e sua aplicação					X
Aprendizagem Esforço dos stakeholders para aprenderem a aplicação					X
Operabilidade Esforço dos stakeholders para operarem e controlarem a operação					X
Aspectos negativos					
O facto de o valor obtido por consenso por um grupo de stakeholders ser sobreposto por um outro grupo posterior, sem que o primeiro seja notificado					
Aspectos positivos					
A integração das atitudes do processo de negociação com o modelo QFD garante a obtenção de resultados fiáveis					

Os dados quantitativos dos formulários (Funcionalidade / Usabilidade) apesar de não serem estatisticamente relevantes, mostram que o sistema foi bem aceite pelos stakeholders.

Os dados qualitativos (Aspectos negativos / positivos) mostram que:

- a arquitectura utilizada no sistema MEG (EXCEL + RTD + DCOM + ADO) terá de ser optimizada para que o tempo de resposta do sistema diminua;
- o sistema auxilia a decisão em grupo;

- não é tão fácil de utilizar como seria suposto à partida (por não informáticos).

Note-se que os dados da experiência 1 se baseiam numa implementação em que os servidores se encontram num computador de recursos reduzidos, o que aumenta os tempos de resposta. Na experiência 2 esses tempos não são notados dado que os servidores e as aplicações cliente se encontram na mesma máquina.

Por outro lado, a sessão introdutória dos conceitos presentes no sistema MEG revelou-se demasiado pormenorizada na experiência 1, expondo demasiado o seu funcionamento e contribuindo para gerar confusão na sua utilização. A exposição tem de se restringir apenas à utilização simples do sistema (tal como na experiência 2).

Utilização do MEGCLIENT

O stakeholder A sugere que é mais intuitivo usar como identificação dos valores de correlação na caixa Value, a identificação {nenhum, fraco, médio, forte} em vez de {0, 1, 3, 9}. Por outro lado, o stakeholder B acha positivo o facto de:

- a árvore da caixa Current Situation fechar por cada alteração que ocorre, obrigando a olhar / ler as alterações efectuadas;
- não mostrar *quantos* já fizeram CONCORDO, ou seja, procura-se o consenso sem que os outros influenciem.

O stakeholder A observou que existe uma situação que poderá suscitar alguma confusão no resultado apresentado: quando todos os stakeholders tem uma posição A_FAVOR relativamente a uma *questão*, se um deles escolher NÃO CONCORDO na *fase pós-negociação*, resulta uma questão rejeitada com todas as posições A_FAVOR.

Uma solução possível será proporcionar ao stakeholder que não concorda, a possibilidade de mudar primeiro o seu valor de correlação para um valor diferente do valor da questão e só depois optar por discordar. O resultado mostrado reflectirá então uma posição CONTRA a questão.

O stakeholder C defende ainda que, apesar de o sistema ser intuitivo, seria preferível disponibilizar a informação da caixa Current Situation em forma gráfica em vez de em árvore. Esta posição está relacionada com o tipo de stakeholder que poderá utilizar o sistema. Por exemplo, para um gestor de topo habituado a ver os resultados de forma gráfica, é mais intuitivo ver que existem N stakeholders a favor / contra um determinado valor do que a disposição em árvore exposta.

Os stakeholders raramente usaram opções do sistema que revelassem atitudes distributivas, sendo encontrados pontos de intersecção com relativa rapidez em ambas as experiências. É de realçar ainda o elevado nível de entusiasmo na utilização do sistema durante as experiências realizadas.

Valores das células da matriz

Um problema levantado pelos stakeholders B e D é de que o sistema não avisa os stakeholders que poderão estar interessados na correlação de uma célula quando esse valor é obtido por consenso por um outro grupo de stakeholders. De facto, sendo S_{CELULA} o conjunto de todos os stakeholders envolvidos na discussão de uma CÉLULA tal que:

$$S_{CELULA} \subset S_{TODOS} \text{ e}$$

corr₁ o valor obtido por consenso por S_{CELULA} para tempo t_1 ,

se em tempo t_2 outro S_{CELULA} obtiver um outro valor $corr_2$, pode suceder que $\exists S_i \in S_{CELULA}$ de t_1 que não se aperceba que o valor por ele concordado, foi alterado.

No limite, a situação é semelhante a S_i colocar um valor em CÉLULA em t_1 e S_j alterar esse valor em t_2 sem que S_i se aperceba, ou seja, na prática, o efeito seria semelhante ao que sucede no sistema VIRTUAL QFD [Herzwurm & Schockert, 1999] descrito no ponto 4.4.4.

A solução passará por assinalar a CÉLULA com uma cor de fundo diferente se já houve um valor obtido anteriormente por consenso por um grupo de stakeholders diferente do actual (a cor de fundo, só se altera na matriz de QFD para os stakeholders que não fizeram parte do novo consenso). Essas células podem ser colocadas com a cor de fundo novamente a branco, se por exemplo o stakeholder a seleccionar, significando que o novo valor foi visto, optando esse stakeholder por iniciar ou não um processo de negociação para essa célula. A técnica de vistas diferentes das matrizes QFD utilizada no sistema CO-DECIDE [Gebhardt et al., 1997] pode ajudar a minimizar o esforço de percepção das células alteradas pela diminuição de células visíveis para um determinado stakeholder.

7.5. Conclusões

O uso do padrão ISO/IEC 9126 veio não só facilitar a elaboração de soluções mas também garantir que as soluções obtidas estão dentro de um padrão de qualidade.

Na perspectiva dos stakeholders, o sistema proporciona suporte à negociação de requisitos. No entanto, existe a necessidade de algum cuidado na sessão explicativa do

sistema MEG. A exposição pormenorizada dos modelos utilizados aumenta a percepção da complexidade dos problemas envolvidos da negociação de requisitos de software por parte dos stakeholders. Essa percepção dificulta o uso do sistema dado que os stakeholders analisam não só o resultado da negociação, mas também a forma de como se chegou a esse resultado, introduzindo questões desnecessárias ao processo. Por outro lado, uma reformulação da visualização dos dados presentes na caixa Current Situation para uma forma mais gráfica será objecto de uma análise futura.

Os dados qualitativos apresentados de seguida são, segundo o stakeholder Z, os mais relevantes das experiências, e que influenciarão alterações futuras do sistema MEG:

Aspectos negativos
<ul style="list-style-type: none">• Sistema lento• Consenso entre um grupo de stakeholders 2 pode sobrepor consenso verificado anteriormente entre um grupo de stakeholders 1• Visualização requer forma mais gráfica
Aspectos positivos
<ul style="list-style-type: none">• Facilidade em encontrar pontos de intersecção

Capítulo 8

Conclusões e trabalho futuro

Pode afirmar-se que o **problema genérico** em foco nesta tese é a negociação de requisitos de software durante a fase de análise de requisitos. Normalmente os projectos de software não correspondem às expectativas dos stakeholders, em grande parte derivado de deficiências ao nível da análise de requisitos. Uma das deficiências prende-se com a resolução de conflitos de interesses entre os vários requisitos de software entre os diversos stakeholders.

Partindo deste problema genérico, procedeu-se ao estudo de diversos **conceitos** envolvidos: negociação, requisitos de software, metodologias de desenvolvimento de software direccionadas para a qualidade, argumentação, consenso e voto maioritário.

Tendo sido analisados diversos sistemas que perseguem os objectivos acima expostos, evidenciaram-se alguns problemas não resolvidos e soluções interessantes em trabalhos relacionados. A tese procurou soluções para os seguintes **problemas**: **(1)** falta de entendimento entre o que *de facto* se pede e o que *se entende* que foi pedido, originando que as soluções apresentadas pelos stakeholders que desenvolvem os sistemas não corresponde às expectativas daqueles que os requerem e **(2)** a negociação de requisitos é normalmente restringida a facilitar apenas a troca de conhecimento entre stakeholders.

As soluções para estes problemas, que são propostas na tese, baseiam-se no desenvolvimento, levado a cabo pelo autor, de um protótipo – sistema MEG. Este

protótipo, descrito em detalhe na dissertação, é constituído por 4 componentes: um **interface** baseado em folha de cálculo, uma **ferramenta de visualização** do estado de negociação, um **servidor de negociação** e um **servidor de sincronização** de dados.

O interface é baseado em Microsoft Excel que permite efectuar automaticamente a *priorização dos requisitos* através do cálculo dos requisitos mais importantes.

A ferramenta de visualização é acedida através do interface Excel, permitindo visualizar e agir segundo uma *estratégia de negociação*.

O servidor de negociação efectua a gestão das mensagens entre stakeholders e a gestão do repositório de dados.

O servidor de sincronização efectua em tempo real a alteração dos dados visíveis no interface Excel.

A **solução** para os problemas (1) e (2) baseia-se na metodologia Desdobramento da Função Qualidade aplicada ao Software (SQFD).

Na metodologia SQFD são estabelecidas ligações entre os requisitos e as soluções através de valores numéricos que expressam os pontos de vista dos stakeholders. Tem-se portanto que a ligação entre *o que* se pretende e *o como* implementar determinada solução que corresponda às expectativas dos stakeholders deriva naturalmente do próprio modelo SQFD, o que justifica a escolha da metodologia como **solução para (1)**.

Por conseguinte, havendo diferentes stakeholders com diferentes pontos de vista, é necessário consenso quanto a um valor único. *O sistema proposto permite evidenciar que a avaliação de um requisito de software, pode ser reduzida ao encontro de um ponto de intersecção de valores, via consenso.*

Por outro lado, no SQFD são usadas matrizes em que se relacionam requisitos / soluções sem mostrar explicitamente os objectivos dos stakeholders. Importa por isso expor as razões que sustentam a escolha de um valor de correlação (*posição*) de um stakeholder face a um valor inserido por outro stakeholder durante o processo de negociação. *O sistema proposto procura assim dar suporte à negociação de correlações*, conforme a figura 8.1.

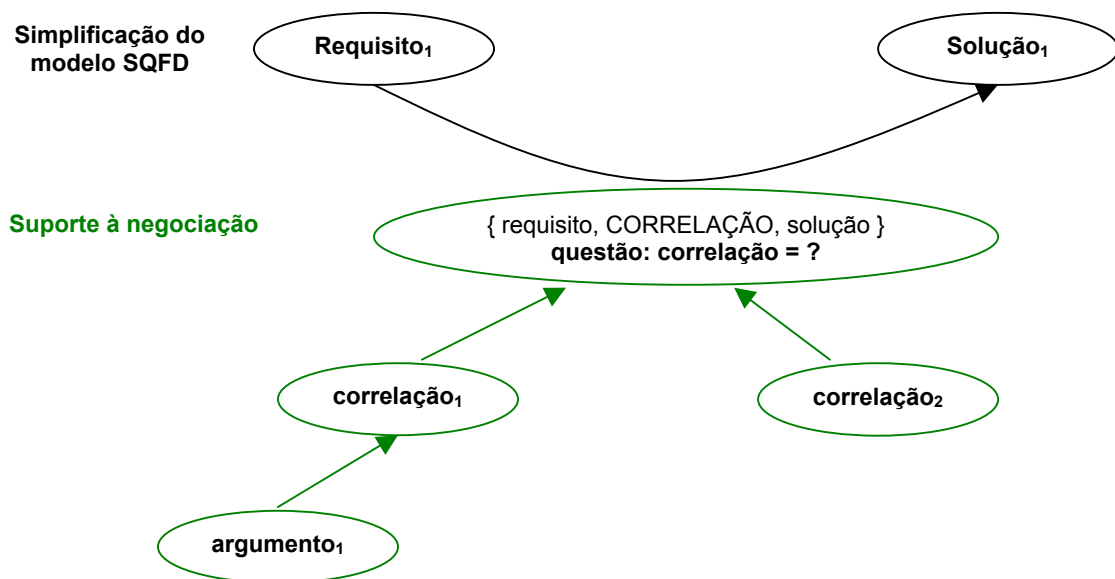


Figura 8.1 – Modelo SQFD com suporte à negociação de correlações

A utilização de sistemas de negociação na vida real tem sido condicionada pela sua dependência no comportamento cooperativo por parte dos stakeholders e na negligência dos processos de dar e receber no processo de negociação. *A solução proposta procura dar suporte ao processo de dar e receber, significando que vai além de simplesmente facilitar a troca de conhecimento, procurando apresentar-se como **solução para (2)**.*

A priorização dos requisitos é um ponto chave do modelo SQFD. *A priorização dos requisitos associado à solução apresentada no capítulo 5, permite uma estratégia de negociação intermédia entre integrativa e distributiva.*

O **principal contributo** deste trabalho para a resolução de conflitos na avaliação de requisitos de software pode ser resumido *à interligação entre o suporte à negociação e o modelo de desenvolvimento de software SQFD.*

Outros contributos passam por permitir esconder a complexidade do processo de negociação na avaliação de requisitos de software mantendo simultaneamente uma reserva de privacidade necessária a um processo de negociação, dado que não são revelados os valores individuais que os stakeholders estão dispostos a aceitar. Procura-se também conduzir os stakeholders a assumirem atitudes integrativas sem limitar as suas opções e propor valores de consenso, requerendo no entanto que os stakeholders explicitamente concordem ou não com esses valores, sendo o consenso por sua vez auxiliado por voto maioritário. Por ultimo, permite focalizar a comunicação, dado que as questões a negociar são específicas.

Uma limitação à solução proposta poderá ser o uso da ontologia conforme referido em Lu et al. (2004) – por um lado, os stakeholders podem não escolher argumentos dado que consideram um aborrecimento tudo o que não é absolutamente necessário e por outro lado, os stakeholders geralmente não sabem o porquê de tomarem uma determinada decisão.

Uma outra limitação à solução proposta está ligada ao facto de ser possível, após uma negociação, o valor negociado ser renegociado por um grupo diferente de stakeholders sem que todos os stakeholders que negociaram da ultima vez sejam avisados, tal como descrito no capítulo 7. Esta falha resulta de se supor condição necessária e suficiente o consenso para um grupo de stakeholders presentes na discussão de uma determinada

célula sem levar em conta que os conjuntos presentes em discussões anteriores podem ser diferentes no decorrer do tempo. O protótipo apresentado encontra-se portanto sujeito a refinamento futuro quanto a este ponto.

Note-se ainda a necessidade de apresentação do sistema proposto a um maior número de stakeholders de forma a permitir uma melhor avaliação¹.

Resta apontar alguns tópicos que não foram analisados em total profundidade nesta dissertação e que, portanto, se consideram objecto de trabalho futuro. Em primeiro lugar, torna-se necessário refinar o processo de renegociação. A solução passa por guardar a informação de todos os stakeholders que participam na discussão de cada célula de forma a inviabilizar o consenso de valores entre grupos diferentes de stakeholders sem que todos os elementos dos grupos sejam notificados dos valores que pretendem ser alterados.

Considera-se como trabalho futuro estender a metodologia proposta a todas as áreas da Casa de Qualidade (HoQ) – e posteriormente ao conjunto de tabelas do modelo SQFD.

Considera-se igualmente como trabalho futuro o estudo de metodologias que facilitem a obtenção de soluções de qualidade nas matrizes SQFD. Este estudo provém da necessidade sentida pelo autor de um método que auxilie a obtenção de soluções que excedam as expectativas dos stakeholders, durante a avaliação preliminar e descrita no capítulo 7, questão esta relacionada com a categorização de soluções.

Por outro lado, considera-se ainda como trabalho futuro o estudo de uma ontologia de suporte aos argumentos.

¹ Tenciona-se avaliar futuras versões do sistema entre stakeholders das diferentes áreas do Centro Nacional de Pensões na avaliação de requisitos de software.

Finalmente, considera-se como trabalho futuro o estudo da apresentação do modelo SQFD para diferentes tipos de stakeholder, num formato que não implique explicitamente tabelas. A abstracção do modelo SQFD em forma de grafo abre a possibilidade a outras representações.

Referências

[Ajenstat & Benchimol, 1999] Jacques Ajenstat, Jaime Benchimol “NSS in Project Management Enterprise Wide Technologies: The Case of PMOffice”, *Proceedings of the 32nd Hawaii International Conference on System Sciences, Hawaii, IEEE Computer Society, HICSS 1999*

[Akao, 1997] Yoji Akao “QFD: Past, Present, and Future”, *International Symposium on QFD – Linkoping, 1997*, http://www.qfdi.org/QFD_History.pdf visitado em 2004/04

[Anderson , 1992] Terry Anderson, “Step Into My Parlor: A Survey of Strategies and Techniques for Effective Negotiation”, *Business Horizons, May-June 1992*

[Belluci & Zleznikov, 1998] Emilia Belluci, John Zleznikov “A Comparative Study of Negotiation Decision Support Systems”, *Proceedings of the 31st Hawaii International Conference on Systems Sciences, Hawaii, IEEE Computer Society, HICSS 1998*

[Behm, 2001] Andreas Behm “Migrating Relational Databases to Object Technology”, *PhD Thesis, Universidade de Zurich, Junho 2001*, http://www.ifi.unizh.ch/ifiadmin/staff/rofrei/Dissertationen/Jahr_2001/thesis_behm.pdf, visitado em 2004/05

[Bohem et al., 1995] Barry Bohem, Prasanta Bose, Ellis Horowitz, Ming June Lee, “Software Requirements Negotiation and Renegotiation Aids: A Theory-W Based Spyral Approach”, *Proceedings of the 17th International Conference on Software Engineering (ICSE-17), Seattle, April 1995*

- [Bohem et al., 2001] Barry Bohem, Paul Grunbacher, Robert O. Briggs, “Developing Groupware for Requirements Negotiation: Lessons Learned”, *IEEE SOFTWARE* May/June 2001
- [Briggs & Grunbacher, 2002] Robert O. Briggs, Paul Grunbacher (2002) “EasyWinWin: Managing Complexity in Requirements Negotiation with GSS”, *Proceedings of the 35th Annual Hawaii International Conference on System Sciences, Hawaii, IEEE Computer Society, HICSS 2002*
- [Bulletinboards, 2004] <http://www.bulletinboards.com/index.cfm>, visitado em 2004/05
- [Chaudhury et al., 1991] Abhijit Chaudhury, H. Raghav Rao, Sukumar Rathnam, “What Can Computer Programs Do To Facilitate Negotiation Processes?”, *ACM SIGOIS Bulletin, Conference proceedings on Organizational computing systems, Volume 12 Issue 2-3, October, 1991*
- [Conklin & Begeman, 1988] Jeff Conklin, Michael L. Begeman “gIBIS: A Hypertext Tool for Exploratory Policy Discussion”, *Proceedings of the 1988 ACM conference on Computer-supported cooperative work Portland, Oregon, United States, Pages:140 - 152, 1988*
- [Cornell, 2001] Paul Cornell (2001) “Building a Real-Time Data Server in Excel 2002”, *Microsoft Corporation, July 2001*, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnexcel2k2/html/odc_xlrtdbuild.asp
- [Costa et al., 2000] Carlos J. Costa, Pedro Antunes, João Ferreira Dias, “GDSS: Limitações e Oportunidades”, *Nova Economia e Tecnologias de Informação: Desafios para Portugal, Luís Valadares Tavares and Manuel João Pereira, Eds. (Seleção de Artigos Da Conferência Sistemas e Tecnologias de Informação: Desafios para o Século*

XXI Realizada Em Lisboa, Outubro 1999). Lisboa: Universidade Católica, 2000, pp. 147-165. (ISBN: 972-54-0019-4).

[Damian et al., 2000] Daniela E. Herlea Damian, Armin Eberlein, Mildred L.G. Shaw, Brian R. Gaines “Group performance and distributed requirements negotiations”, *March 22, 2000* http://pharos.cpsc.ucalgary.ca/Dienst/UI/2.0/Describe/ncstrl.ucalgary_cs/2000-650-02 visitado em 2004/01

[Gebhardt et al., 1997] Michael Gebhardt, Matthias Jarke, Stephan Jacobs, “A Toolkit for Negotiation Support Interfaces to Multi-Dimensional Data”, *Proceedings of ACM SIGMOD Conference, Tucson, Arizona, 1997*

[Getsch & Cornell, 2001] Tim Getsch, Paul Cornell “Real-Time Data: Frequently Asked Questions”, *Microsoft Corporation, July 2001*

[Gilb, 2003a] Tom Gilb, “Towards the Engineering of Requirements”, www.result-planning.com visitado em 2003/01

[Gilb, 2003b] Tom Gilb, “Competitive Engineering”, www.result-planning.com visitado em 2003/01

[Grunbacher & Briggs, 2001] Paul Grunbacher, Robert O. Briggs (2001) “Surfacing Tactic Knowledge in Requirements Negotiation: Experiences using EasyWinWin”, *Proceedings of the 34th Annual Hawaii International Conference on System Sciences, Hawaii, IEEE Computer Society, HICSS 2001*

[Grunbacher et al., 2003] Paul Grunbacher, Sabine Koszegi, Michael Halling, Stefan Biffel, “Integrating Concepts of Negotiation Theory into Software Requirements Negotiation”, *Working Paper OP2003-09, 2003*,

http://gerda.univie.ac.at/orgplan/forschung/working_papers/OP2003-09.pdf visitado em 2004/03

[Haag et al., 1996] Stephen Haag, M. K. Raja, L. L. Schkade “Quality Function Deployment Usage in Software Development”, *Communications of the ACM, (Vol.39 No.1) January 1996*

[Herzwurm et al., 1997] Georg Herzwurm, Werner Mellis, Sixten Schockert, Claudius Weinberger “Customer Satisfaction Survey of QFD Software Tools”, *A. Gustafsson, B. Bergman, F. Ekdahl (Hrsg.): Proceedings of the Third Annual International QFD Symposium in Linköping, Sweden, Volume 1, S. 309-323, October 1-2, 1997*

[Herzwurm et al., 1999] Georg Herzwurm, Sixten Schockert, Werner Mellis “Higher Customer Satisfaction with Prioritizing and Focused Software Quality Function Deployment” *Proceedings of the Sixth European Conference on Software Quality in Vienna, April 12-16, 1999. Wien 1999*

[Herzwurm et al., 2002] Georg Herzwurm, Sixten Schockert, Ulrike Dowie, Michael Breidung “Requirements Engineering for Mobile Business Applications”, *Proceedings of the First International Conference on Mobile Business, July 8-9, 2002 in Athens, Greece 2002*

[Herzwurm & Schockert, 1999] Georg Herzwurm, Sixten Schockert “Virtual Product Development – Using the Internet as a Communication Platform for QFD” *Proceedings of the Fifth International Symposium on Quality Function Deployment and the First Brazilian Conference on Management of Product Development, S. 83-91, August 24-26, 1999 in Belo Horizonte, Brazil, Belo Horizonte, 1999*

[Hierholzer et al., 1998] Andreas Hierholzer, Georg Herzwurm, Harald Schlang “Applying QFD for Software Process Improvement at SAP AG, Walldorf, Germany”,

Ross L. Chapman, Robert Hunt (eds.): *Proceedings of the World Innovation and Strategy Conference in Sydney, Australia, S.85-95, August 2-5, 1998*

[**Horowitz et al., 1999**] Ellis Horowitz, Joo H. Lee, June Sup Lee “WinWin: a System for Negotiating Requirements”, *Proceedings of the 21st international conference on Software engineering, Los Angeles, California, United States, p.646-649, May 16-22, 1999*

[**Hwang & Lin, 1987**] Ching-Lai Hwang, Ming-Jeng Lin “Group Decision Making under Multiple Criteria – Methods an Applications”, *Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, 1987*

[**ISO/IEC9126, 1991**] ISO 9126: The Standard of Reference, Information technology - Software Product Evaluation - Quality characteristics and guidelines for their use – 1991, <http://www.cse.dcu.ie/essiscope/sm2/9126ref.html> visitado em 2004/04

[**Jacobs & Kethers, 1994**] Stephan Jacobs, Stefanie Kethers "Improving Communication and Decision Making within Quality Function Deployment", *1st Conference on Concurrent Engineering Research and Application, CERA, Pittsburgh, August 1994*

[**Jarke et al., 1987**] Matthias Jarke, M. Tawfik Jelassi, Melvin F. Shakun “MEDIATOR: Towards a negotiation support system” , *European Journal of Operational Research 31, 1987*

[**Karacapilidis & Papadias, 1997**] Nikos Karacapilidis, Dimitris Papadias “Computer Supported Argumentation and Collaborative Decision Making: The Hermes System”, <http://www.mech.upatras.gr/~nikos/papers/hermes.pdf> visitado em 2003/12

-
- [Karacapilidis & Papadias, 1997a]** Nikos Karacapilidis, Dimitris Papadias “A framework for Group Decision Support Systems: Combining AI tools and OR techniques”, *European Journal of Operational Research* 103: 373-388, 1997
- [Karacapilidis et al., 1998]** Nikos Karacapilidis, Dimitris Papadias, Costas P.Pappis “Modeling Negotiations in Group Decision Support Systems”, in *T. Stewart and R.v.d Honert (eds.), Trends in Multicriteria Decision Making - Proceedings of the 13th International Conference on Multiple Criteria Decision Making, Cape Town, South Africa, Springer-Verlag, Lecture Notes in Economics and Mathematical Systems, pp. 163-176, Vol. 465, 1998*
- [Kersten, 1987]** Gregory E. Kersten “On Two Roles Decision Support Systems Can Play In Negotiations”, *Information Processing & Management Vol. 23 Pergamon Journals, 1987*
- [Knogstie, 1999]** John Knogstie “Using Quality Function Deployment in Software Requirements Specification”, www.ifi.uib.no/konf/refsq99/papers/krogstie.rtf visitado em 2003/12
- [Korhonen et al., 1992]** Pekka Korhonen, Helbert Moskowitz, Jyrki Wallenius “Multiple criteria decision support – A review”, *European Journal of Operational Research* 63, 1992
- [Lewicki & Litterer, 1985]** Roy J. Lewicki, Joseph A. Litterer “Negotiation”, *IRWIN, 1985*
- [Lin et al., 1996]** Jinxin Lin, Mark S. Fox, Taner Bilgic “A Requirement Ontology for Engineering Design”, *Concurrent Engineering: Research and Applications, Vol.4, No.3, 279-291, 1996*

[Lu et al., 2004] S.C-Y.Lu, F.Udwadia, B.Bulket, J.Cai “Design Rationale for Collaborative Engineering Design in the Socio-Technical Framework”, *Engineering School, University of Southern California, Los Angeles, CA 90089-1111, United States*, <http://impact.usc.edu/cerl/Task2/task2-2-3/DesignRationale.pdf>, visitado em 2004/01

[Madu, 1994] Christian N. Madu “A Quality Confidence Procedure for GDSS Application in Multicriteria Decision Making”, *IIE Transactions Vol. 26 Number 3, May 1994*

[Maine, 2004] “Consensus-Based Stakeholder Processes: What is a consensus process?” http://www.maine.gov/consensus/ppcm_consensus_home.htm visitado em 2004/01

[Microsoft, 2000] “Use Dcomcnfg for a Visual Basic DCOM Client/Server Application”, *Microsoft Knowledge Base Article – 183607, Microsoft Corporation, 2000*

[Microsoft, 2001] “Use an Excel RTD Server with DCOM”, *Microsoft Knowledge Base Article – 285888, Microsoft Corporation, 2001*

[Microsoft, 2003] “Configure DCOM for Visual Basic Using DCOMCNFG.EXE”, *Microsoft Knowledge Base Article – 183607, Microsoft Corporation, 2003*

[Microsoft, 2004] “ActiveX Data Objects (ADO)”, *MSDN Library*, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ado270/htm/mdmscsection1_ado.asp visitado em 2004/01

[Mazur, 1993] Glenn H. Mazur, “QFD for Service Industries – From Voice of Customer to Task Deployment”, *The Fifth Symposium on Quality Function Deployment, Novi, Michigan, United States, June, 1993*

- [Nour, 2003] Philip Nour “Ontology-based Retrieval of Software Engineering Experiences”, *master thesis, University of Calgary, 2003*,
<http://sern.ucalgary.ca/~milos/papers/2003/Nour2003.pdf> visitado em 2004/05
- [Nuseibeh & Easterbrook, 2000] Bashar Nuseibeh, Steve Easterbrook “Requirements Engineering: A Roadmap”, *Proceedings of International Conference on Software Engineering (ICSE-2000), The Future of Software Engineering, A. Finkelstein (ed.), Limerick, Ireland, ACM Press, 4-11, June 2000*
- [Park et al., 1999] Jung-Won Park, Daniel Port, Barry Boehm, Hoh In “Supporting Distributed Collaborative Prioritization for WinWin Requirements Capture and Negotiations”, *International Workshop on Process support for Distributed Team-Based Software Development (PDTSD'99) World MultiConference SCI/ISAS'99, Orlando, United States, August, 1999*
- [Peixoto & Carpinetti, 1998] Manoel Peixoto, Luiz Carpinetti “Aplicação de QFD integrando o modelo de Akao e o modelo QFD-estendido”, *Gestão e Produção, v.5, n.3, p.221-238, 1998*
- [Planetsourcecode, 2003] “Basic TCP/IP Multi-Client Echo Server”,
<http://www.planetsourcecode.com/vb/scripts/ShowCode.asp?txtCodeId=39661&lngWId=1>
- [Ramesh & Bui, 1999] Balasubramaniam Ramesh, Tung Bui “Negotiation in Network Based Requirements Analysis”, *Proceedings of the 32nd Hawaii International Conference on Systems Sciences, Hawaii, IEEE Computer Society, HICSS 1999*
- [Ramesh & Mohan, 2001] Balasubramaniam Ramesh, Kannan Mohan “Integrating Group Decision and Negotiation Support Systems with Work Processes”, *Proceedings*

of the 34th Annual Hawaii International Conference on System Sciences, Hawaii, IEEE Computer Society, HICSS 2001

[Ramires & Antunes, 2004] João Ramires, Pedro Antunes “Negociação de Requisitos no modelo SQFD – Software Quality Function Deployment”, *Primeira Conferência Nacional em Interacção Pessoa-Máquina, Julho 12-14, 2004 Lisboa, Portugal, 2004*

[Reich, 1996] Yoram Reich “AI-supported Quality Function Deployment”, *Artificial Intelligence in Economics and Management (P. Ein-Dor, ed.), pp. 91--106, Kluwer, 1996*

[Robinson & Volkov, 1998] William Robinson, Vecheslav Volkov “Supporting the Negotiation Lifecycle”, *Communications of the ACM, 41(5), 1998*

[Seedsforchange, 2004] “Consensus Decision Making”,
<http://seedsforchange.org.uk/res/consens.html> visitado em 2004/01

[Sommerville & Sawyer, 1997] Ian Sommerville, Pete Sawyer “Requirements Engineering, A Good Practice Guide”, *Jonh Wiley and Sons, 1997*

[Stylianou et al., 1997] Antonis C. Stylianou, Ram L. Kumar, Moutaz J. Khouja (1997) “A Total Quality Management-Based Systems Development Process”, *The DATA BASE for Advances in Information Systems – Summer 1997 (Vol.28, No.3)*

[Zultner, 1993] Richard E. Zultner “TQM for Technical Teams”, *Communications of the ACM, October 1993 (Vol.38 No.10)*