

Negociação de Requisitos no modelo SQFD - Software Quality Function Deployment

João Ramires
Centro Nacional de Pensões
Av. República 102, 3º Lisboa
joao.j.ramires@seg-social.pt

Pedro Antunes
LaSIGE – Faculdade de Ciências da
Universidade de Lisboa
Campo Grande
paa@di.fc.ul.pt

Sumário

Este artigo descreve um sistema para a resolução de conflitos na avaliação de requisitos de software. A originalidade da abordagem está na interligação entre suporte à negociação e o modelo de desenvolvimento de software SQFD (Software Quality Function Deployment).

Palavras-chave

GDSS, NSS, SQFD, Win-Win.

1. INTRODUÇÃO

Apresenta-se neste artigo uma nova abordagem para a resolução do problema bem conhecido do conflito na avaliação de requisitos de software. A abordagem proposta recorre a um sistema que interliga um grupo de pessoas – utilizadores, programadores, chefes de projecto, gestores de topo, etc, normalmente denominados por *stakeholders* – e que suporta o recurso a técnicas de negociação na resolução dos conflitos que surgem na avaliação dos vários requisitos de software. O protótipo desenvolvido, denominado MegSystem, é simultaneamente um Sistema de Suporte à Decisão em Grupo (GDSS) e um Sistema de Suporte à Negociação (NSS) assente no modelo de desenvolvimento de software SQFD (Software Quality Function Deployment), fomentando uma perspectiva de negociação integrativa (Win-Win).

O artigo está estruturado da seguinte forma. Apresentam-se em primeiro lugar algumas definições, de modo a contextualizar o trabalho desenvolvido. Em seguida descreve-se a integração do modelo SQFD com o modelo de negociação. Finalmente descrevem-se alguns pormenores de implementação do sistema.

1.1 Definições

Negociação

A negociação pode ser definida como um “processo social básico utilizado para resolver conflitos” [Lewicki85]. Este processo envolve a procura, identificação e escolha de uma alternativa (ponto de intersecção) dentro de um conjunto de alternativas possíveis e aceitáveis pelos *stakeholders*. Este conjunto de alternativas pode sofrer alterações, expandindo ou contraindo até ser encontrado um ponto de intersecção [Kersten87]. O processo de negociação pode ainda ser dividido segundo duas estratégias [Chaudhury91]: a) integrativa: tenta-se chegar a acordo de forma inventiva, cooperativa e na busca persistente de ganhos conjuntos, recorrendo-se a uma comunicação aberta e partilha de informação – ambas as partes ganham ou pelo menos nenhuma perde (Win-Win); b) distributiva: baseada em posições duras e inflexíveis 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 (Win-Lose).

As atitudes que os indivíduos podem tomar na negociação são essenciais para se entender as diferenças entre as estratégias integrativa e distributiva (Figura 2). Existem cinco atitudes que os participantes podem seguir numa situação de conflito [Lewicki85]: a) COMPETITION (competição): tentar convencer a outra parte a aceitar uma posição que favorece apenas os próprios interesses; b) ACCOMODATION (acomodação): em contraste à competição, uma estratégia de acomodação ajuda a outra parte a atingir os seus próprios objectivos. Em vez de focar nas próprias necessidades, o foco é nas necessidades dos outros. Os negociadores podem acomodar-se por várias razões: acabar rapidamente a negociação (pressão do tempo), deixar a outra parte satisfeita para pedir algo mais tarde, ou porque as questões são mais importantes para a outra parte; c) COMPROMISE (compromisso): quando os negociadores repartem entre si os ganhos. Nem um nem outro ganha. Em vez disso, cada lado concorda numa divisão da questão de maneira a que cada um obtenha parte da solução pretendida – mas não tudo. As partes chegam a um compromisso quando, vendo as questões em termos de ganhos fixos, podem repartir as diferenças de modo a que cada um fique com uma parte. É um resultado satisfatório mas não óptimo, dado que cada parte não consegue alcançar todos os seus objectivos; d) COLLABORATION (colaboração): quando as partes colaboram para maximizar os resultados da sua união. É um processo muito similar à resolução de problemas em grupo, já que as partes compreendem que partilham um problema em termos de alvos e interesses partilhados. Cada parte expõe as suas necessidades, e ambas trabalham para inventar opções que preencham essas necessidades de maneira optimizada; e) AVOIDANCE (inacção): as partes podem decidir simplesmente evitar o conflito. Se um negociador não luta pelos seus próprios objectivos nem mostra preocupação pelos objectivos dos outros, ele está a evitar o conflito. A inacção 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 as intenções do opositor. Esta estratégia pode ser usada quando, por

exemplo, o negociador teme negociar, as questões são vistas como insignificantes ou se pretende protelar a negociação.

Requisitos

Os requisitos de software são definidos durante as fases iniciais do processo de desenvolvimento de software. Definem-se como “descrições de como o sistema se deve comportar ou uma propriedade / atributo do sistema” [Sommerville97], balizando o processo de desenvolvimento. A avaliação de requisitos de software consistirá no confronto dos objectivos dos *stakeholders* perante um conjunto de constrangimentos – duração, custo, etc. Os conflitos na avaliação de requisitos de software resultam de perspectivas, interpretações e posições diversas dos *stakeholders*.

NSS e GDSS

A distinção entre Sistemas de Suporte à Negociação (NSS) e Sistemas de Suporte à Decisão em Grupo (GDSS), como tipos específicos de sistemas cooperativos de suporte à decisão é ambígua. Ambos 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 na distinção entre os dois tipos de sistemas está relacionado com a própria definição de negociação [Grunbacher03]: as tarefas de negociação são sempre caracterizadas por um conflito entre os interesses das partes envolvidas, enquanto que nas tarefas de decisão em grupo tal não será explicitamente o caso. Na decisão em grupo assume-se, na maior parte do tempo, uma atitude integrativa dos membros do grupo, onde o conflito é resolvido pela partilha de informação. Ao invés, na negociação, apesar de existirem interesses comuns, existem normalmente atitudes distributivas que inviabilizam a representação do problema de forma conjunta.

A aproximação dominante nos GDSS que lidam com requisitos de software assume que os *stakeholders* seguem sempre atitudes integrativas: os objectivos / preferências individuais podem ser agregados numa representação conjunta do problema e os conflitos são resolvidos pela facilitação do acesso à informação e 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 resolvidos pela partilha de informação. Nestes casos uma estratégia interessante será integrar os conceitos de GDSS e NSS.

QFD e SQFD

Apesar de as metodologias tradicionais de desenvolvimento de software reconhecerem a importância dos utilizadores, elas não fornecem métodos formais que transformem os requisitos de qualidade dos utilizadores em especificações de desenho do sistema. Por outro lado, estas metodologias não mantêm rigorosamente a atenção nos requisitos de qualidade definidos pelos utilizadores nas fases mais avançadas do desenvolvimento de software [Stylianou97].

Uma das técnicas utilizadas por organizações que implementam programas de Gestão de Qualidade Total (Total Quality Management, TQM) [Zultner93] para manter o foco nos requisitos de qualidade dos clientes durante todo o processo de design, produção e entrega dos produtos denomina-se Quality Function Deployment (QFD) [Zultner93]. A técnica QFD tem sido igualmente aplicada ao processo de desenvolvimento de software – Software QFD (SQFD) [Haag96], onde a qualidade exigida é cada vez maior, sendo que com o QFD se consegue [Akao97]: a) identificar essa qualidade; b) um método para a construção de sistemas que assegurem essa qualidade.

2. NEGOCIAÇÃO NO MODELO SQFD

No SQFD são utilizadas matrizes em que se comparam os requisitos com as soluções na forma de correlações (*corr*). Para estas correlações são possíveis os valores *none*, *weak*, *medium* e *strong* que equivalem aos valores numéricos 0, 1, 3 e 9 respectivamente. As correlações identificam a preferência de uma solução que satisfaz o requisito. Por exemplo, na figura 4, o requisito “Write emails fast/easily” correlacionado com a solução “Enter email via voice” através de um valor 9, sugere que esta solução é eficaz segundo o ponto de vista de um stakeholder. No conjunto dos *stakeholders* S podem existir no entanto, vários valores possíveis para *corr* o que origina conflito de interesses, exigindo portanto negociação (figura 1). A resolução de conflitos

de *corr* afecta directamente a avaliação de quais são os requisitos mais importantes, com consequências significativas no processo de desenvolvimento de software.

O Virtual QFD [Herzwurm99] permite aos *stakeholders* alterarem os valores das correlações das matrizes pelos seus próprios valores, exigindo ao grupo, uma contínua verificação das alterações efectuadas. O Co-Decide [Gebhardt97] permite a inserção de valores individuais dos *stakeholders*, assinalando as células da matriz que estão em conflito. O MegSystem reflete não só as situações de conflito, como também auxilia a resolução de conflitos de *corr* entre *stakeholders*, obtendo um valor por consenso.

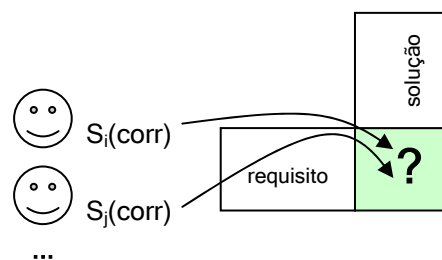


Figura 1 – Conflito entre stakeholders para um valor de correlação

O MegSystem propõe ainda um valor baseado em voto maioritário e avaliação do comportamento distributivo dos *stakeholders*. Na descrição da abordagem proposta pelo MegSystem vai apenas ser considerado o contexto da matriz Software House of Quality (sw-HoQ) do modelo SQFD, dado que o processo a aplicar nas restantes matrizes é análogo.

Seja S_{ALL} o conjunto de todos os *stakeholders* presentes no sistema e S_{CELL} o conjunto de todos os *stakeholders* envolvidos na discussão de uma célula CELL tal que $S_{CELL} \subset S_{ALL}$, S_i e S_j *stakeholders* $\in S_{CELL}$ e S_{OTHERS} o conjunto tal que $S_{CELL} = S_i \cup S_{OTHERS}$. Vai ser assumido que o processo se desenrola numa CELL, em que CELL pode estar num estado de Equilíbrio E ou de Negociação T:

$$CELL = \{ \{E | T\}, \{corr, S_{CELL}\} \} \text{ com } E = \{a, b\}, T = \{n, o, p, x, z\} \text{ e } corr = \{NONE | WEAK | MEDIUM | STRONG\}$$

O conjunto de Atitudes possíveis na negociação são:

ATTITUDE = {COMPETITION |
ACCOMODATION | AVOIDANCE |
COLLABORATION | COMPROMISE}

As atitudes podem ainda ser divididas em
Distributivas e Integrativas (figura 2):

ATTITUDE_{DIST} = {COMPETITION |
AVOIDANCE_{DIST}}

ATTITUDE_{INTE} = {COLLABORATION |
ACCOMODATION | COMPROMISE |
AVOIDANCE_{INTE}}

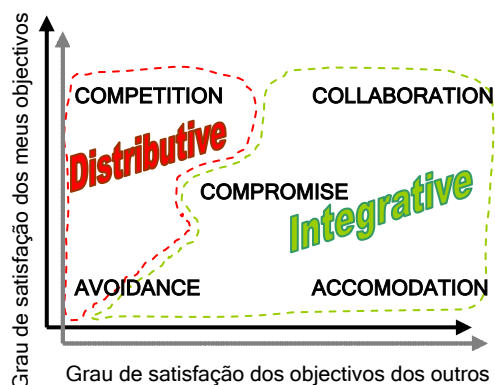


Figura 2 - Atitudes Distributivas e Integrativas

A separação da atitude AVOIDANCE deriva da definição encontrada em [Lewicki85]: a) AVOIDANCE_{INTE} significa que o *stakeholder* se retira da situação de conflito, ou seja, retira todos os seus valores de CELL permitindo à outra parte fazer o que quiser; b) AVOIDANCE_{DIST} cria uma dependência na negociação, frustrando os opositores. No primeiro caso, é normalmente seguida uma estratégia integrativa para os casos de negociação considerados insignificantes, enquanto no segundo é seguida uma estratégia distributiva, onde se pretende por exemplo, demorar a negociação para benefício próprio.

O suporte à Argumentação (AS) é composto por:

AS = { Issue, { Position(S_i), { Argument(S_i) } } } ;
com: Issue = { ISSUE | ISSUE_PROPOSED |
ISSUE_STAKEHOLDER_FIRM }; Position(S_i)
= { NONE | AGAINST | IN_FAVOR };
Argument(S_i) = { ARG } ∈ ONTOLOGY.

Entenda-se: ISSUE como o primeiro valor inserido para *corr*; ISSUE_PROPOSED como o valor proposto para *corr* pelo sistema face ao contexto; ISSUE_STAKEHOLDER_FIRM

como o valor proposto para *corr* por um *stakeholder* com uma atitude explícita de COMPETITION; e ONTOLOGY como o conjunto de palavras reservadas que fazem sentido num determinado contexto. As relações entre entidades de AS são: Issue HAS Positions; Position SUPORTED_BY Arguments.

Os valores seleccionados por cada *stakeholder* não são conhecidos dos restantes – apenas o sistema os conhece. O sistema facilita a tomada de posição pública de um *stakeholder* perante a Issue: o sistema assume implicitamente e divulga a Position do *stakeholder*, quer relativamente à ISSUE, quer relativamente à ISSUE_PROPOSED baseada nos valores de todos.

A estratégia integrativa assenta em [Anderson92]: a) separar as pessoas dos problemas; b) focar nos interesses, não nas posições; c) criar opções de ganho mútuo; d) insistir no uso de critérios objectivos.

A solução proposta procura promover a estratégia integrativa, optando-se por: a) esconder a identidade dos *stakeholders* quando têm comportamentos integrativos. Ao esconder a identidade os intervenientes atacam o problema em vez de se atacarem mutuamente, indo ao encontro dos dois primeiros pontos do parágrafo anterior; b) propôr um valor (ISSUE_PROPOSED) que, baseado em voto maioritário, cria uma opção para ganho mútuo, assente num critério de preferência e no comportamento dos *stakeholders* de forma a promover justiça (dois últimos pontos do parágrafo anterior).

A estratégia integrativa é condicionada pelo grau de desconfiança das estratégia usadas pelos oponentes, originando que muitos ganhos conjuntos não sejam alcançados por permanecerem escondidos. Procura-se combater isso 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 expôr demasiado “o jogo.”

Uma posição dura e inflexível, onde não existe preocupação pelas necessidades dos outros, é normalmente o comportamento usado na

estratégia distributiva. Esta atitude firme é suportada na solução proposta, mas penalizada de modo a procurar conduzir os *stakeholders* a uma estratégia integrativa. Aliás, a atitude $AVOIDANCE_{DIST}$ é igualmente penalizada pela mesma razão. Segue-se a descrição dos estados possíveis para CELL – estado de Equilíbrio E ou de Negociação T (consultar Figura 3).

Estado E(a)

Representa o estado inicial, ou seja, a CELL encontra-se vazia. Passa ao estado E(b) com a inserção de um valor x por S_i em CELL. $E(a) \rightarrow x \rightarrow E(b)$.

Estado E(b)

Sendo inserido um valor *corr* y por outro *stakeholder* S_j , se o novo valor for igual ao que se encontra em CELL, continua no estado E(b). Senão, passa ao estado T(n). $E(b) \rightarrow y = x \rightarrow E(b)$; $E(b) \rightarrow y \neq x \rightarrow T(n)$.

Estado T(n)

Existem pelo menos dois valores diferentes em CELL, originando um estado de conflito. Para resolver o conflito, é necessário os *stakeholders* tomarem uma Atitude: $T(n) \rightarrow ATTITUDE_{INTE} \rightarrow T(o)$; $T(n) \rightarrow COMPETITION \rightarrow T(x)$; $T(n) \rightarrow AVOIDANCE_{DIST} \rightarrow T(z)$.

Estado T(o)

Tem-se que os *stakeholders* alteraram os valores de *corr* numa atitude Integrativa. Se existe um ponto de intersecção entre os intervalos definidos que origina uma Position IN_FAVOR para a Issue em S_{CELL} , então $T(o) \rightarrow IN_FAVOR(S_i)$ and $IN_FAVOR(S_{OTHERS}) \rightarrow T(p)$; $T(o) \rightarrow T(n)$ caso contrário.

Estado T(p)

Verifica-se se é possível o consenso entre *stakeholders*. Se todos AGREE passa ao estado de Equilíbrio E(b). Se existe um NOT_AGREE passa ao estado de Negociação T(n). $T(p) \rightarrow \forall AGREE \rightarrow E(b)$; $T(p) \rightarrow \exists NOT_AGREE \rightarrow T(n)$.

Estado T(x)

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

$T(n)$. $T(x) \rightarrow AGREE \rightarrow T(p)$; $T(x) \rightarrow NOT_AGREE \rightarrow T(n)$.

Estado T(z)

Se um *stakeholder* S_i teve uma atitude $AVOIDANCE_{DIST}$ a partir do estado de negociação T(n), não é possível passar ao estado T(p). A passagem ao processo de consenso é assim bloqueada o que não torna possível a obtenção de um valor para CELL. $T(z) \rightarrow ATTITUDE_{INTE} \rightarrow T(z)$. Apenas S_i pode retirar o bloqueio.

Segue-se a descrição de AS em cada estado T.

AS para o estado T(n)

No início de T(n), o sistema gera automaticamente o seguinte AS:

$AS = \{ ISSUE, \{ Position(S_i) = IN_FAVOR, \{ \} \} \{ Position(S_j) = AGAINST, \{ \} \} \}$

Sendo o valor y de S_j diferente do valor x colocado por S_i em primeiro lugar na CELL, e que por conseguinte deu origem à ISSUE, tem-se que a Position de S_j é AGAINST, e a Position de S_i será IN_FAVOR ao próprio valor inserido. Neste estado inicial não é necessário que os *stakeholders* insiram os seus argumentos.

AS para o estado T(o)

O sistema tenta propôr um valor para Issue (ISSUE_PROPOSED) baseado em voto maioritário e no contexto das atitudes dos *stakeholders*. Por cada valor escolhido é atribuído um peso $w = 1$. Se o *stakeholder* tiver tido n atitudes COMPETITION a partir de T(n), o peso dessa alternativa é penalizado $n \cdot 10^{-3}$ vezes ou seja $w = 1 - (n \cdot 10^{-3})$. Se o *stakeholder* tiver tido n atitudes $AVOIDANCE_{DIST}$, o peso dessa alternativa é penalizado $2n \cdot 10^{-3}$ vezes. Por exemplo, para as condições: Valor(S_i) = 3, com uma atitude $AVOIDANCE_{DIST}$ verificada anteriormente; Valor(S_j) = 9 tem-se que:

ISSUE = 3

Position = IN_FAVOR

Position = AGAINST

ISSUE_PROPOSED = 9

Position = AGAINST

Position = IN_FAVOR

$w_i(3) = 1 - (2 * 1 * 0,001) = 0,098$; $w_j(9) = 1$;
 $MAX(w_i, w_j) = 1 \Rightarrow ISSUE_PROPOSED = 9$

Note-se que depois de ser obtida a *ISSUE_PROPOSED*, é aplicado o mesmo critério das *Positions* utilizado na *ISSUE* relativamente aos valores de cada *stakeholder* face à *Issue* correspondente.

Apesar de o sistema se basear no voto maioritário para propôr 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 $MAX(w_i, w_j)$ não fosse único, o sistema não poderia propôr um valor.

Os *stakeholders* poderão ainda indicar os Argumentos de forma a explicar as razões das Posições assumidas. Baseado numa ontologia, AS permite focalizar a comunicação do grupo em itens específicos evitando a verbalização.

AS para o estado T(p)

Pode-se chegar a T(p) de três formas:

- ISSUE
 Position = IN_FAVOR, S_{CELL}
- Ou ISSUE_PROPOSED
 Position = IN_FAVOR, S_{CELL}
- Ou ISSUE_STAKEHOLDER_FIRM

A última *Issue* é conseguida através de uma atitude *COMPETITION* originada em T(n).

AS mostra o valor da *Issue* que originou T(p), com o fim de ser aprovado por todos e passar-se ao estado E(b) com *Issue* = *ISSUE*.

AS para o estado T(x)

AS apenas mostra:

ISSUE_STAKEHOLDER_FIRM.

AS para o estado T(z)

Só é permitido mostrar em AS {*ISSUE*, *ISSUE_PROPOSED*}. Note-se que mesmo que se verifique *Position = IN_FAVOR, S_{CELL}*, mantém-se o mesmo AS por impossibilidade de se passar a T(p).

3. ESQUEMA DE INTERACÇÃO

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

correção 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).

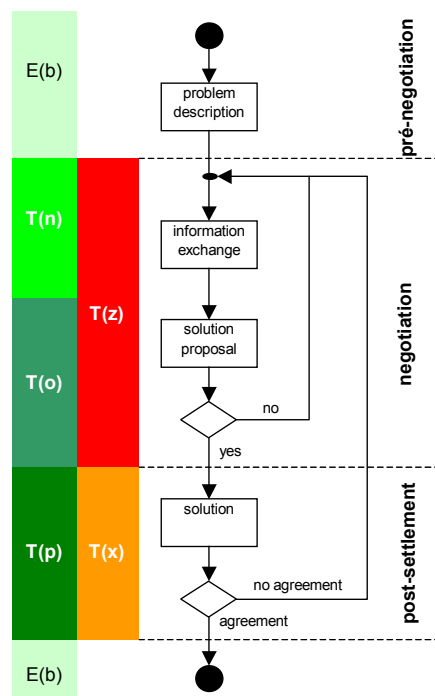


Figura 3 - Processo de Negociação

A negociação consiste em 3 fases: *pre-negotiation*, *negotiation* e *post-settlement* (Figura 3). A *pre-negotiation* apresenta o problema, isto é, prepara os pontos a negociar para a fase seguinte. Na fase de *negotiation* são trocados argumentos que sustentam as posições e novas propostas / argumentos entre *stakeholders*, via sistema. Se se chega a uma solução (caminho *yes* na Figura 3), é necessário obter o consenso dos *stakeholders* envolvidos na discussão (caso contrário continua a troca de propostas / argumentos). Se na fase *post-settlement* se chega a um consenso quanto à solução proposta (caminho *agreement* na Figura 3), o valor encontrado passa a ser a solução (caso contrário continua a troca de propostas / 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.

Em seguida descrevem-se os diversos estados do sistema e correspondentes interações dos *stakeholders*.

Estado 1 – Descrição do problema

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

Estado 2 – Partilha de informação

Associado ao estado T(n): cada *stakeholder* pode inserir mais do que um valor, significando o alargamento de valores possíveis de modo a encontrar um ponto de intersecção de interesses. Representa as atitudes implícitas ∈ ATTITUDE_{INTE} consoante os valores / intervalos tendam para o valor da Issue. É ainda

possível a inserção de argumentos que sustentam as Position. A inserção de valores dá origem à passagem ao estado T(o). Uma atitude COMPETITION remete-nos 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 AVOIDANCE_{DIST} impossibilita a passagem à fase de *post-settlement*, ou seja, não permite que se chegue a consenso (apenas permite atitudes ∈ ATTITUDE_{INTE}).

Estado 3 – Proposta de uma solução

Associado ao estado T(o): o sistema apresenta uma solução (estado T(p)) quando todas as posições dos *stakeholders* envolvidos forem IN_FAVOR relativamente à Issue em discussão (ISSUE) ou proposta pelo sistema (ISSUE_PROPOSED). Se existe uma Position AGAINST, passa-se para o estado T(n).

O valor de ISSUE_PROPOSED é baseado nos valores introduzidos pelos *stakeholders*. Quando

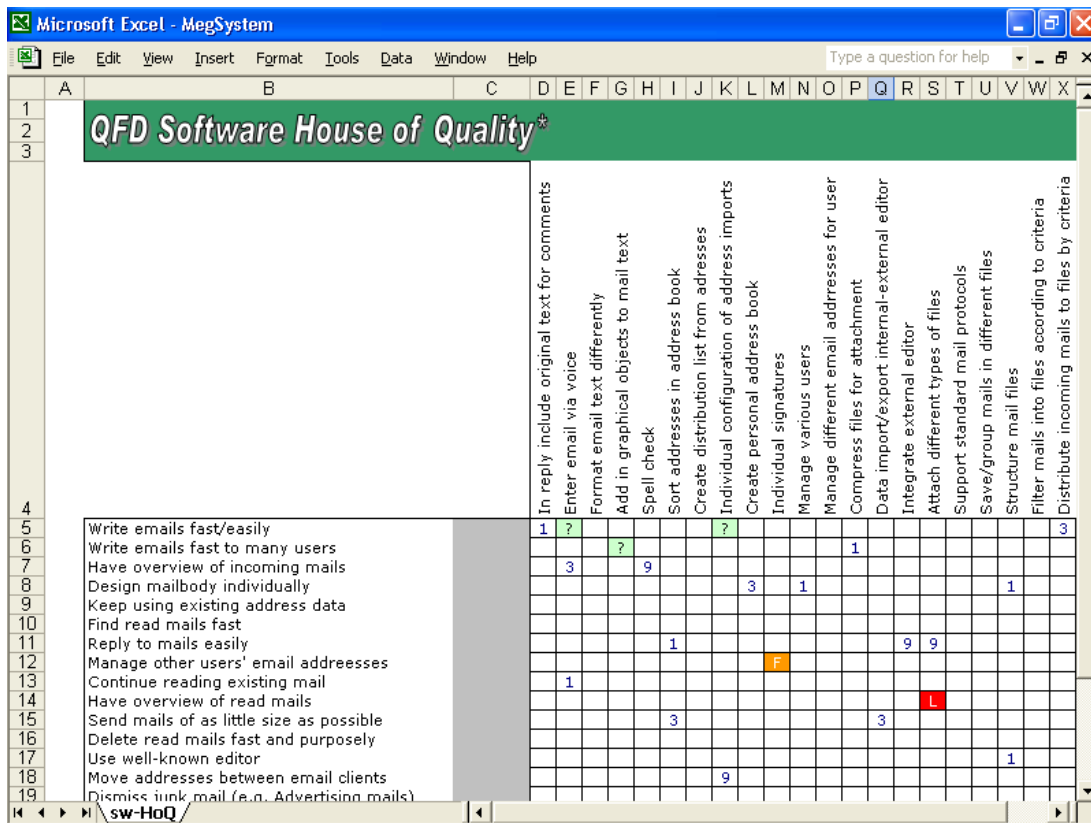


Figura 4 – Tabela SQFD [Herzwurm02] no MegSystem

as posições dos *stakeholders* mudam, este valor é actualizado automaticamente.

Estado 4 - Solução

O sistema chega a uma solução em duas situações: a) todas as posições são a favor de uma Issue; b) existe um e um só S_i com uma atitude COMPETITIVE em $T(n)$, sendo o valor proposto o correspondente ao valor de S_i .

A solução apresentada é sujeita a consenso. Ou todos concordam na solução a que se chegou por consenso ou todos concordam com uma posição. Se a solução é aprovada, passa novamente a um estado E(b).

4. INTERACÇÃO COM O SISTEMA

Cada utilizador é confrontado com uma folha de cálculo Microsoft Excel 2002 que resume a sw-HoQ (Figura 4), apresentando as alterações dos valores das correlações inseridos pelos outros utilizadores em tempo-real. Os símbolos {?, F, L} significam respectivamente que (1) se está num processo de negociação; (2) alguém optou por FIRM; e (3) alguém optou por LOCK à célula.

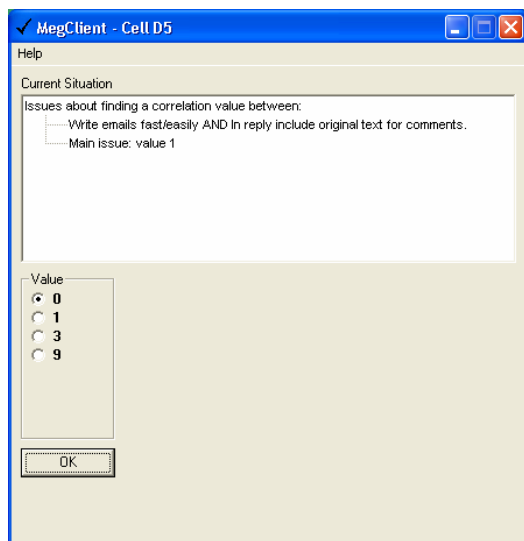


Figura 5 – Formulário – *pre-negotiation*

A manipulação de valores da sw-HoQ é feita a partir da ferramenta MegClient, invocada por duplo *click* sobre as células da matriz. Um *stakeholder* que não tem valores gravados numa célula, pode inserir apenas um e um só valor e ver o resultado na forma de posições (atitude COMPETITION ou ACCOMODATION). Na

Figura 5 vemos que a célula já continha o valor 1. Na fase de negociação, importa perceber as razões das várias posições dos *stakeholders*, pelo que poderá ser necessário inserir argumentos. Na Figura 6 mostra que a inserção do valor 3 com um argumento, gerou uma alteração na caixa Current Situation. As atitudes $ATTITUDE_{INTE}$ são obtidas implicitamente a partir dos valores inseridos (nesta fase podem ser inseridos mais do que um simultaneamente); as atitudes COMPETITION e $AVOIDANCE_{DIST}$ são expressas respectivamente através das opções FIRM e LOCK disponíveis em My Preferences, significando respectivamente que: (1) “mantenho a posição porque é absolutamente imprescindível que assim seja” e (2) “não quero negociar com ninguém”. A oferta de opções relacionadas com atitudes $ATTITUDE_{DIST}$ em My Preferences destina-se a torná-las menos acessíveis. Passa à situação *post-settlement* quando se escolhe a opção FIRM ou quando a solução proposta tem a posição IN_FAVOR para S_{CELL} . Se nesta fase se acrescentar o valor 1, a posição passa a ser também IN_FAVOR, pelo que se obtém o formulário da Figura 7 com *issue* = 1 para ser sujeito a consenso.

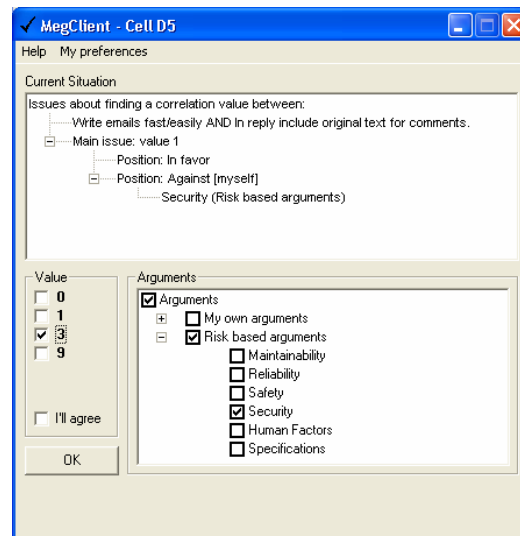


Figura 6 – Formulário – *negotiation*

A opção I'll agree (figura 6) seleccionada permite definir a priori que em caso de passagem à situação *post-settlement*, se concorda com os valores em discussão desde

que sejam iguais aos seleccionados em Value (diminui o tempo do processo de consenso). Em *post-settlement* se todos escolherem AGREE passa-se a um estado E(b) e esse valor é colocado na célula da matriz. A caixa Current Situation da Figura 7 mostra a natureza do valor sujeito a consenso: se o valor proposto é de autoria de um *stakeholder* por atitude FIRM ou por serem todos a favor do valor da *issue* ou do valor proposto pelo sistema (aparece apenas a *issue* da qual se pretende o consenso).



Figura 7 - Formulário – *post-settlement*

A opção LOCK inibe o processo de negociação (ver Warning na Figura 8). Excesso de atitudes distributivas são mal interpretadas num ambiente que promove atitudes integrativas (o número de vezes de uso dos botões LOCK ou FIRM não só é revelado como também interfere no valor proposto pelo sistema, Figura 9).

5. IMPLEMENTAÇÃO

Os programas Excel são sincronizados através de um servidor (MegServer) usando TCP/IP e tecnologia RTD [Cornell01]. O MegSystem (servidor e clientes) foi desenvolvido em VB 6. As mensagens e o repositório de dados são em formato XML.

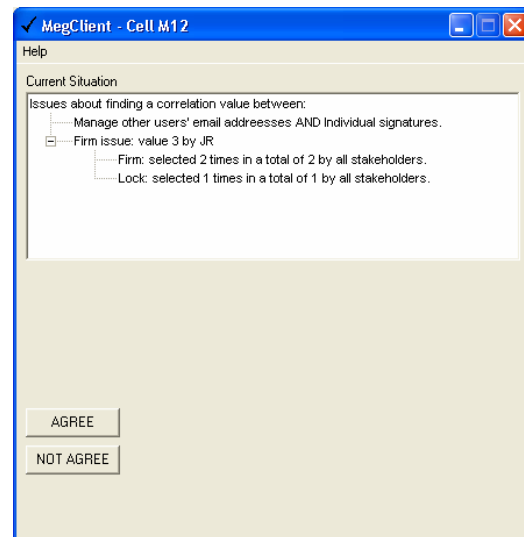


Figura 9 - Formulário – *post-settlement* com FIRM

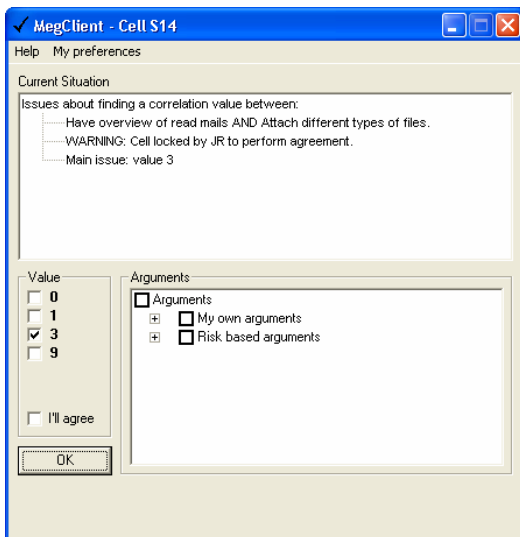


Figura 8 - Formulário – *negotiation* com LOCK

6. CONCLUSÃO

O MegSystem procura dar suporte à resolução de conflitos na avaliação de requisitos de software. Assente no modelo SQFD, permite que os *stakeholders* cheguem a acordo relativamente a valores de correlação entre requisitos e soluções, valores esses que representam perspectivas, interpretações e posições diversas dos *stakeholders*.

Recorrendo à ferramenta de inserção de valores MegClient, esconde-se a complexidade do processo de avaliação de requisitos de software. Sem limitar as opções dos utilizadores, o sistema tenta conduzi-los numa estratégia integrativa, em três passos fundamentais [Lewicki85]: a) identificação do problema; b) geração de soluções alternativas para o

problema; c) escolha de uma solução específica de entre as alternativas possíveis.

Apesar de propor valores de consenso, o sistema requer que os utilizadores explicitamente concordem ou não com esses valores. O sistema também nunca fecha uma negociação, sendo sempre possível a um utilizador mudar de ideias e alterar uma célula. Finalmente, o sistema não divulga todas as alternativas que os utilizadores estão dispostos a aceitar, mantendo a reserva de privacidade necessária a um processo de negociação.

O MegSystem encontra-se em avaliação no Núcleo de Análise de Sistemas de Informação da Unidade de Informática do Centro Nacional de Pensões.

7. AGRADECIMENTOS

Os autores agradecem os comentários do grupo HCIM (Human Computer Interaction and Multimedia Group) da Faculdade de Ciências da Universidade de Lisboa.

8. REFERÊNCIAS

[Anderson92] Terry Anderson, “Step Into My Parlor: A Survey of Strategies and Techniques for Effective Negotiation”, *Business Horizons*, May-June 1992

[Akao97] Yoji Akao “QFD: Past, Present, and Future”, International Symposium on QFD – Linkoping, 1997

[Chaudhury91] Abhijit Chaudhury, H. Raghav Rao, Sukumar Rathnam, “What Can Computer Programs Do To Facilitate Negotiation Processes?”, *ACM SIGOIS Bulletin*, v12(2-3), 1991

[Cornell01] Paul Cornell (2001) “Building a Real-Time Data Server in Excel 2002”, *Microsoft Corporation*, July 2001

[Gebhardt97] Michael Gebhardt, Matthias Jarke, Stephan Jacobs, “A Toolkit for Negotiation Support Interfaces to Multi-

Dimensional Data”, *Proceedings of ACM SIGMOD Conference, Tucson, Arizona, 1997*

[Grunbacher03] Paul Grunbacher, Sabine Koszegi, Michael Halling, Stefan Biffel, “Integrating Concepts of Negotiation Theory into Software Requirements Negotiation”, *OP2003-09*, 2003

[Haag96] Stephen Haag, M. K. Raja, L. L. Schkade “Quality Function Deployment Usage in Software Development”, *Communications of the ACM, January 1996 (Vol.39 No.1)*

[Herzwurm99] 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, August 24-26, 1999 in Belo Horizonte, Brazil, Belo Horizonte 1999, S. 83-91*

[Herzwurm02] 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*

[Lewicki85] Roy J. Lewicki, Joseph A. Litterer “Negotiation”, *IRWIN*, 1985

[Kersten87] Gregory E. Kersten “On Two Roles Decision Support Systems Can Play In Negotiations”, *Information Processing & Management Vol. 23 Pergamon Journals*, 1987

[Sommerville97] Ian Sommerville, Pete Sawyer “Requirements Engineering, A Good Practice Guide”, *JONH WILEY & SONS*, 1997

[Stylianou97] 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)*

[Zultner93] Richard E. Zultner “TQM for Technical Teams”, *Communications of the ACM, October 1993 (Vol.38 No.10)*