

Sumário

1 Introdução da Monografia.....	2
2 Fundamentos.....	5
2.1 Requisitos.....	5
2.1.1 Tipos de Requisitos.....	5
2.2 Engenharia de Requisitos	6
2.2.1 Produção de Requisitos.....	7
2.2.2 Gerência de Requisitos.....	7
2.3 O léxico ampliado da linguagem.....	8
2.4 O RNF-Framework.....	10
2.4.1 Geração de Grafos de RNfs.....	12
Alteração para Facilitar a Rastreabilidade Reversa.....	12
Alterações para Facilitar a Integração das Visões	13
Funcional e Não - Funcional	13
Construção do Grafo de RNfs.....	16
2.4.2 Nossa Proposta de Alteração na Simbologia do Grafo.....	18
2.5 Análise de Negócio: Integração dos RNfs aos Modelos de Processos de Negócio.....	19
2.5.1 Modelos de negócio.....	19
2.5.2 Modelos de Processos de negócio.....	20
2.5.3 A importância do uso de modelagem de processos de negócio para a Engenharia de Requisitos.....	20
2.5.4 A Abordagem BPMNRNF	20
2.6 O SQFD.....	21
3 Negociação de Requisitos de Software	21
3.1 Aspectos Conceituais	21
3.1.1 Conceitos Fundamentais de Negociação.....	21
3.2 Conflitos: tipos, causas	22
3.2.1 Conflitos de Requisitos.....	23
3.3 Abordagens de Negociação.....	23
3.4 Aspectos de Relacionais e Substantivos.....	23
3.4.1 O processo de Negociação	23
3.4.1.1 Sistema de apoio a resolução de conflitos	24
Negociação no modelo SQFD.....	24
Máquina de Estados.....	26
Processo de Negociação de Requisitos	27
4 Estratégia de Engenharia de Requisitos.....	34
5 Projeto Arquitetural.....	34
6 Análise da Viabilidade.....	34
7 Resultados.....	34
8 Conclusões.....	34
9 Trabalhos Futuros.....	34

1 Introdução da Monografia

Este trabalho monográfico apresenta os resultados do projeto de pesquisa **Negociação de Requisitos: uma análise da viabilidade de integração entre técnicas que estimulam a participação do cliente**, que tinha como objetivo fazer um levantamento e estudo sobre métodos, técnicas ou estratégias que facilitassem a negociação de requisitos de software e verificar sobre a possibilidade de formulação de uma abordagem metodológica de negociação que pudesse ser desenvolvida e aprimorada para que pudesse ser adaptada utilizada pelas empresas de Teresina-PI. Procuramos demonstrar argumentos, práticas, subpráticas e constatações de autores que estudam o assunto e assuntos correlatos bem como as observações e constatações tiradas a partir da análise e de reflexões feitas durante o estudo com o fim de demonstrar a importância do tema Negociação de Requisitos. O projeto também tinha como objetivos específicos: procurar por compatibilidades/incompatibilidades entre as propostas selecionadas, procurar por pontos carentes de adaptações para torná-las compatíveis entre si, bem como identificar os pontos onde tais adaptações seriam mais fáceis e mais trabalhosas.

Estudamos trabalhos de autores como Ramires (2004), Camacho (2005), Cysneiros (2001), Couto e Martins (2008), com a intenção de conseguir uma base de cunho específico e prático para a argumentação a ser usada na monografia e também para delimitação do tema. Os trabalhos de Carvalho (2001), Kantorski e Kronth (), Laurindo e Moraes (2004), Gusmão (2008), Germoglio (2008), foram analisados com o fim de se extrair relações com os trabalhos dos autores anteriormente citados e enriquecer a pesquisa bem como: verificar tendências, contextualização de situações, relações entre as abordagens, em fim, verificar pontos ou aspectos interessantes e em comum. Estudou-se também trabalhos mais gerais como Carvalhal (2006), Leite (1994), Lopes (2002), Pressman (2001), PMBOK (2004), Lee, Cha e Kwon (1998), Swebok (2004), SOMMERVILLE (1997), entre outros, pelos motivos anteriormente citados e, também, para reforçar a argumentação e alinhar as ideias específicas com ideias gerais, amplamente aceitas e já legitimadas no meio da engenharia de software, e mais especificamente na engenharia de requisitos.

Durante a fase de Requisitos (elicitação, documentação e análise) podem ser identificados conflitos entre os requisitos do cliente. O cliente torna-se “obrigado” a negociação junto aos engenheiros sobre a satisfação dos requisitos que envolvem aspectos, muitas vezes, de difícil compreensão para o cliente. Isso pode dificultar a boa participação do cliente durante o processo de desenvolvimento de software.

O cliente tendo pela frente uma situação de negociação envolvendo aspectos que não conhece necessita de tempo para o entendimento do objeto de negociação, dos riscos por trás da não resolução dos conflitos, para que aí sim possa tomar decisões sobre requisitos de forma lúcida.

Geralmente o cliente solicita alterações nos requisitos quando o software já passou pela programação e uma amostra já foi apresentada. Consertar um defeito relacionado a requisito depois da implementação em uma linguagem de programação é sem dúvida mais dispendioso que fazer isso durante fase de requisitos.

Este trabalho aborda o problema da não preparação dos processos de desenvolvimento de software para as necessidades e limitações dos clientes quanto a visualização das consequências dos impactos entre requisitos. Faz um estudo da viabilidade de formulação de uma metodologia (método + estratégia de uso) que posiciona o projeto de desenvolvimento de software de forma a suprir as necessidades e limitações dos cliente no que diz respeito ao entendimento dos efeitos positivo e negativos que os requisitos podem causar uns nos outros e nas fases seguintes do desenvolvimento.

Com a evolução acelerada da informática o mercado de trabalho vem se tornando cada vez mais competitivo e exigente. Devido a isso a demanda por softwares que cada vez mais satisfaçam as necessidade dos clientes e que, ao mesmo tempo custem menos, sejam mais complexos e mais rápidos de serem desenvolvidos aumenta rapidamente.

Ao longo da história da Engenharia de software tem se observado que a importância dada ao requisitos vem aumentando. Porém, o número de trabalhos feitos em que o foco maior se concentra nos requisitos funcionais é muito alto se comparado com os que , além dos funcionais, consideram os atributos de qualidade. Muito avanço já se conseguiu com pesquisas onde o foco maior estava em requisitos funcionais: desenvolvimento de software dirigido por casos de uso, especificação de requisitos com caso de uso, especificação de casos de uso, análise de pontos de função, etc...

A busca pela qualidade faz com que haja uma tendência natural em se preocupar com atributos de qualidade e restrições. Alguns poucos estudos já consideram que a qualidade que o cliente espera que o seu software tenha não é determinada somente com base em requisitos funcionais :

“ Mesmo com a execução rigorosa das fases de engenharia de requisitos e de análise, a experiência na construção de sistemas mostra que ainda existe uma lacuna de informações a serem especificadas para prosseguir com a fase de projeto. Isto significa que especificar e modelar o que o sistema deve fazer não é suficiente para saber como o sistema deve ser estruturado e organizado para satisfazer os requisitos funcionais e os atributos de qualidade.”. (VAROTO, 2002)

Nesse cenário, de preocupação em como o software deverá realizar suas funcionalidades, temos possibilidade de identificação de possíveis impactos positivos ou negativos em outros requisitos que determinam a qualidade do software. No caso dos impactos negativos surge a necessidade de negociação para que se escolha um requisito em detrimento de outro ou que se contente com a satisfação parcial de um, ou para que se encontre alternativas que satisfaçam a necessidade das parte sem que hajam conflitos. Mas isso se agrava com o fato do cliente (leigo) não conseguir entender e, por consequência, não conseguí visualizar esse impacto tornando a negociação algo extremamente difícil para as partes. Isso se deve ao caráter abstrato que o software possui.

O sucesso que é esperado de um projeto de software é altamente relacionado e dependente de uma arquitetura bem definida e apropriada. Mas para isso é necessário que haja sucesso nas negociações, ou seja, que elas sejam realizadas com eficiência e efetividade.

Então para que se tenha uma arquitetura consistente e confiável primeiro é necessário que se tenha o mínimo de requisitos (selecionados lucidamente) necessários à definição da arquitetura e isto depende muito do resultado de negociações. Uma arquitetura mal definida afeta a comunicação da equipe de desenvolvimento, o crescimento do software e decisões sobre as estruturas que formarão o sistema. Afeta também, a estrutura global de controle que será usada, protocolos de comunicação, sincronização e acesso a dados, atribuição de funcionalidades a elementos do sistema ou ainda sobre distribuição física dos elementos do sistema. Além disso, o problema envolverá decisões que impactarão no comportamento do sistema em termos de escala e desempenho. GARLAN e SHAW (apud Germoglio, 2009). Mas antes da melhora na comunicação da equipe de desenvolvimento do software temos que a comunicação nas negociações iniciais, que ocorrem entre cliente e equipe de engenharia de software, gerentes de projeto etc., precisam ser melhoradas, bem como a manipulação ágil e representação dos objetos de negociação, que no caso deste trabalho é os requisitos funcionais e não funcionais.

Mudanças nos requisitos tornam o processo de desenvolvimento cada vez mais caros na medida em que as fases do processo de desenvolvimento vão se passando. Correções nas fases iniciais são mais baratas. E isso obriga os clientes e engenheiros de software a anteciparem as tomadas de decisões quanto a fatores críticos do projeto.

Por isso se faz necessária a negociação de requisitos conflitantes para tomada antecipada de decisões necessárias, afim de que se chegue a um consenso comum entre clientes e engenheiros de software sobre os requisitos principais e, também para que se tenha o mínimo de informação para que se possa prosseguir com a fase de projeto e de desenvolvimento.

Assim qualquer esforço no sentido de diminuir tempo de familiarização do cliente com aspectos inerentes aos impactos dos requisitos durante a negociação dos requisitos com o objetivo de diminuir os riscos de mudanças em fase avançadas do processo de desenvolvimento é bem vindo. Uma idéia nesse sentido seria criar um mecanismo que facilite identificar os requisitos críticos para o sucesso do projeto e facilitar o consenso entre cliente e engenheiros sobre o escopo de requisitos. E que ao mesmo tempo garanta que as principais tomadas de decisão sejam feitas nas fases iniciais, reduzindo assim os riscos referente a custo, tempo e qualidade.

A necessidade constante de sempre produzir softwares que cada vez mais custem menos, que sejam produzidos mais rapidamente, e de acordo com as expectativas (e necessidades) dos stakeholders (ou seja, que tenham qualidade), revelam outras necessidades mais específicas dos processo de desenvolvimento. A necessidade de que o cliente seja mais participativo no processo de desenvolvimento de software é uma delas. Para a elicitação, análise e negociação de requisitos sua presença é fundamental.

Embora Elicitação, Análise e negociação de requisitos sejam conceitos que sempre andam juntos durante o processo de desenvolvimento (Sommerville, Sawyer, 1997). Esse trabalho foca-se na negociação de requisitos. Mas não sugere separação entre elas, por fazerem parte de um processo em espiral (Sommerville, Sawyer, 1997).

A negociação é um instrumento e, ao mesmo tempo, uma necessidade presente durante o processo de desenvolvimento de software. É essencial e de uso inevitável em qualquer projeto que envolva pelo menos 2 partes. Ela também tem suas necessidades

próprias. E busca-se produzir instrumentos e desenvolver técnicas, métodos , estratégias etc. com o objetivo de satisfazer suas necessidades.

Nas negociações pode-se verificar o grau de contentamento, necessidade, interesse... atuais no qual o cliente se encontra.

O capítulo 2 deste trabalho apresenta alguns conceitos básicos de Engenharia de Requisitos necessários ao entendimento deste trabalho.

O capítulo 3 discorre sobre Negociação de uma forma geral e básica.

O capítulo 4, intitulado de “Estratégias de Engenharia de Requisitos”, traz algumas propostas estratégicas de autores brasileiros para o trato com requisitos.

O capítulo 5 contextualiza o capítulo 3 para a área de Engenharia de Software , delimitando o assunto hierarquicamente, passando por Engenharia de Requisitos até chegar em Negociação de Requisitos.

O capítulo 6 fala sobre o projeto arquitetural de software .

O capítulo 7 “ Análise da Viabilidade ” é uma discussão a respeito da viabilidade e desenvolvimento de uma metodologia de engenharia de requisitos que favoreça as negociações da qualidade do software.

O capítulo 8 apresenta as conclusões, de forma resumida, tiradas durante a pesquisa bem como sugestões de trabalhos futuros que podem ser feitos apartir desta monografia.

2 Fundamentos

Apresentaremos neste capítulo alguns conceitos fundamentais para esta monografia que estão relacionados com o conceito de requisitos e engenharia de requisitos. Dentre os conceitos estão os de: requisitos, análise de negócio, léxico ampliado da linguagem, RNF-Framework, SQFD e engenharia de requisitos.

2.1 Requisitos

Segundo o Swebok (2004), um requisito de software é uma propriedade que deve ser exibida, a fim de resolver algum problema no mundo real. O Guia refere-se a requisitos de "software", porque é preocupado com os problemas a serem tratados pelo software. A seguir veremos os tipos e alguns exemplos de cada tipo.

2.1.1 Tipos de Requisitos

Os tipos de requisitos que são considerados na literatura de engenharia de software são os requisitos: funcionais, não-funcionais, e de domínio.

Os requisitos funcionais são aqueles que representam o que o software deve fazer. São exemplos de requisitos funcionais, (Spínola, e Ávila, 2007):

- O software deve permitir o cadastro de clientes;
- O software deve permitir a geração de relatórios sobre o desempenho de vendas no semestre;
- O software deve permitir o pagamento de das compras através de cartão de crédito;

Os requisitos não-funcionais dizem respeito à qualidade com que o software desempenha suas funcionalidades. Esses requisitos colocam restrições no sistema, ou seja expressam condições que o software deve atender. Spínola e Ávila citam exemplos:

- “O software deve ser compatível com os browsers IE (versão 5.0 ou superior) e firefox(1.0 ou superior)”;
- “o software deve garantir que o tempo de retorno das consultas não seja maior que 5 segundos”;

Já os de domínio têm origem no domínio da aplicação. Descrevem as características da aplicação e qualidades que expressão o domínio. Podem, segundo Spínola e Ávila, ser requisitos funcionais novos, restrições sobre requisitos, ou computações específicas. Exemplos:

- “O cálculo da média final de cada aluno é dado pela formula: $(Nota1 * 2 + Nota2 * 3) / 5$ ”;
- “Um aluno pode se matricular em uma disciplina desde que ele tenha sido aprovado nas disciplinas consideradas pré-requisitos”;

O projeto que não atende aos requisitos dos clientes e aos requisitos de processo muito provavelmente irá fracassar. Existem muitos desafios quando se trata de requisito, portanto é necessário que haja uma abordagem sistemática dos requisitos para que se diminua os riscos que os requisitos representam.

2.2 Engenharia de Requisitos

Segundo Soares, o processo de engenharia de requisitos é composto por quatro atividades de alto nível (Soares apud Wikipédia, 2010):

1. Identificação.
2. Análise e negociação.
3. Especificação e documentação.
4. Validação.

Outra atividade que pode ser adicionada a esse processo é a gestão dos requisitos (manutenção ou change management).

Segundo Spínola e Ávila (2007, p. 49), o termo Engenharia de requisitos pode ser definida como: ” termo usado para descrever as atividades relacionadas à produção (levantamento, registro, validação e verificação) e gerência (controle de mudanças, gerência de configuração, rastreabilidade, gerência de qualidade dos requisitos) de requisitos” .

A Figura xxxx representa essa definição.

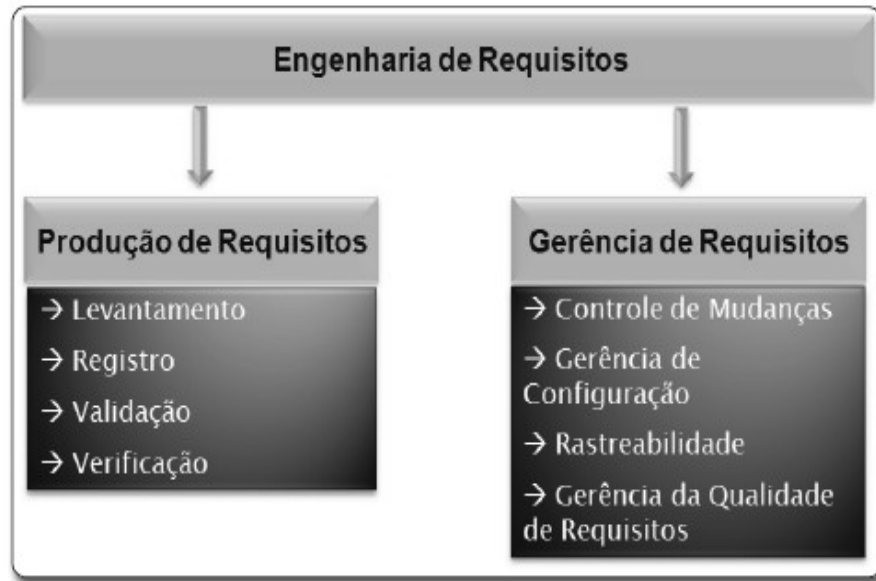


Figura xxxx: Engenharia de Requisitos

Fonte: SPÍNOLA E ÁVILA (2007:48)

2.2.1 Produção de Requisitos

Quatro atividades são consideradas atividades base relacionadas com a Produção de Requisitos: levantamento, registro, verificação e validação.

O *levantamento* está relacionado com a obtenção dos requisitos do software. Nesta atividade os “engenheiros e analistas trabalham com clientes e usuários finais para descobrir o problema a ser resolvido, os serviços do sistema, o desempenho necessário, restrições de hardware e outras informações” (SPÍNOLA E ÁVILA, 2007: 49).

Na atividade de *registro* que ocorre a documentação dos requisitos. Essa documentação servirá de base para as outras fases do processo de desenvolvimento. Geralmente produz-se um documento de especificação de requisitos.

Na atividade de *verificação* ocorre o exame da especificação de requisitos do software, o objetivo é encontrar e corrigir ambiguidades, inconsistências ou omissões antes que a fase de definição de requisitos acabe.

Por fim, segundo Spínola e Ávila, na atividade de validação o objetivo é conseguir o aceite do cliente sob determinado artefato. Na engenharia de requisitos validar é aprovar requisitos de software.

2.2.2 Gerência de Requisitos

A “atividade de administrar os requisitos ao longo do tempo” (SPÍNOLA e ÁVILA, 2007, p. 57) é denominada gerência de requisitos. Dentre as atividade de gerência de requisitos que devem ser levadas em consideração estão: Controle de Mudanças, Gerência de Configuração, Rastreabilidade e Gerência de Qualidade de Requisitos.

Controle de Mudanças

As mudanças nos requisitos ocorrem e é “praticamente inevitável” que elas ocorram. Portanto, deve-se sempre trabalhar com a idéia de mudanças nos requisitos irão ocorrer e que isto é natural. Obviamente que limites devem ser estabelecidos e que deve haver controle sobre tais mudanças, sob o risco de que ,se não houver, o projeto possa fracassar.

A “baseline”(uma pratica bastante usada) permite que se possa diferenciar o que era o requisito original, o que foi introduzido e o que foi descartado. O uso de sistemas para fazer este controle provê mais exatidão à baseline e rapidez aos manipuladores das baselines.

O estabelecimento de processo de controle de mudanças também é importante. Uma atividade importante é a análise das solicitações de mudanças para que se possa aceitar ou rejeitar a mudança.

(SPÍNOLA E ÁVILA, 2007) sugerem passos a serem seguidos para um processo de controle de mudança:

- “Checar validade da solicitação de mudança”;
- “Identificar os requisitos diretamente afetados com a mudança”;
- “Identificar dependências entre requisitos para buscar os requisitos afetados indiretamente”;
- “Assegurar com solicitante a mudança a ser realizada”;
- “Estimar custos da mudança”;
- “Obter acordo com usuário sobre o custo da mudança”.

Gerência de Configuração

A atividade de Gerência de Configuração, segundo Presman

“é o conjunto de atividades projetadas para controlar as mudanças pela identificação dos produtos do trabalho que serão alterados, estabelecendo um relacionamento entre eles, definindo o mecanismo para o gerenciamento de diferentes versões destes produtos, controlando as mudanças impostas, e auditando e relatando as mudanças realizadas.”

Dentre os benefícios da Gerência de Configuração podemos citar: Redução de defeitos; Diminuição de custos para manutenção; Melhora no gerenciamento do projeto

(rastreadabilidade, reproduzibilidade; confiabilidade; visibilidade das mudanças); Melhora na produtividade do desenvolvimento; Melhor retorno do investimento etc...

Podemos citar como objetivos da Gerência de Configuração:

- Definir políticas para controle de versões, garantindo a consistência dos artefatos produzidos;
- Definir procedimentos para solicitações de mudanças;
- Administrar e auditar o ambiente de GC;
- Facilitar e automatizar a geração de build do sistema;
- Facilitar a geração de Release.

Precebe-se aqui o quão importante é a gerência de configuração para a gerência de projetos de software e engenharia de requisitos. Oportunamente, ela é muito importante para as negociações no processo de desenvolvimento do software, pois provê agilidade no trato dos itens de configuração. E isto fortalece a comunicação e, conseqüentemente, o relacionamento entre os stakeholders do projeto.

Rastreabilidade

G

Gerência de Requisitos

R

A gerência de requisitos provê uma infra-estrutura necessária às atividades de verificação, o que possibilita a investigação da qualidade dos requisitos (SPÍNOLA e ÁVILA, 2007).

A gerência de requisitos bem feita é um pré-requisito para uma negociação bem feita.

2.3 O léxico ampliado da linguagem

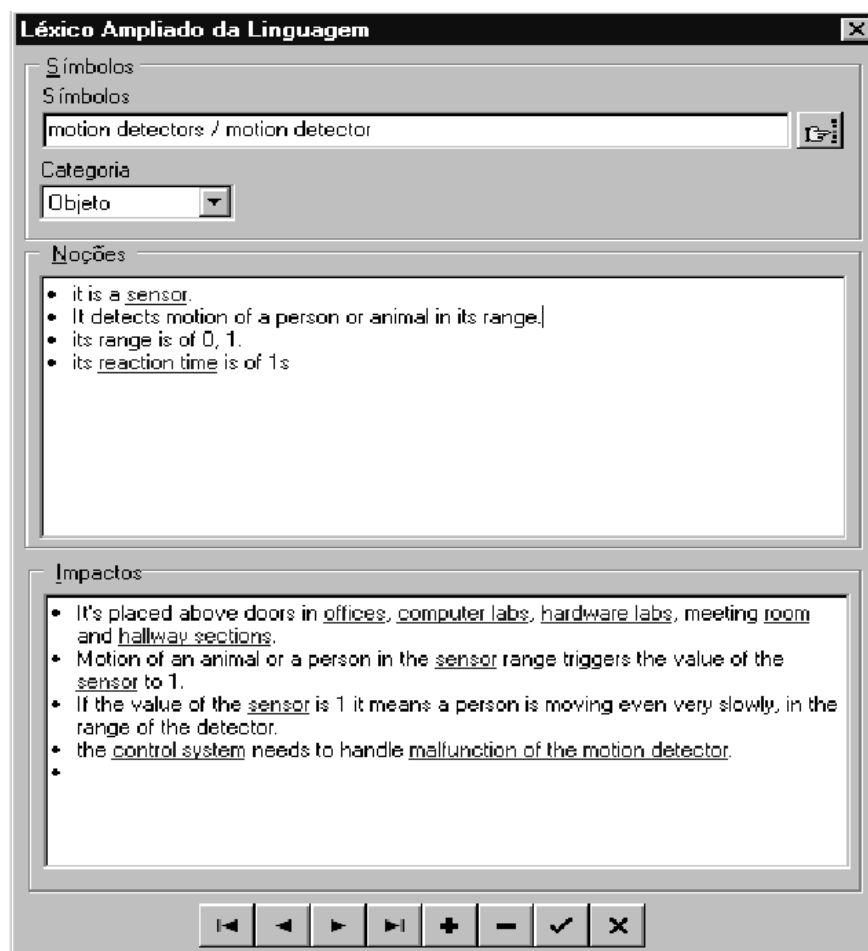
O conceito de *léxico*, de uma forma geral, é o acervo de palavras de um determinado idioma. No contexto do ciclo de desenvolvimento do software e organização para a qual o software será ou esta sendo desenvolvido, o “léxico” pode ser entendido como sendo o acervo de palavras do “idioma de uma organização”. O Léxico Ampliado da Linguagem tem como objetivo registrar os termos e não se preocupar com a funcionalidade do Universo De Informações (UDI). De forma resumida o Léxico Ampliado da Linguagem (LAL) do UDI é o conjunto de termos praticados pelos atores do UDI.

Sua composição é feita de “entradas” relacionadas a “símbolos da linguagem do UDI”. Cada entrada é associada a um símbolo do UDI. Os símbolos possuem “sinônimos” e são descritos por “noções” e “impactos”. As noções expressam a denotação do símbolo, ou seja, o significado mais as relações fundamentais de existência do símbolo com outros símbolos. E os impactos demonstram a conotação, ou seja expressa os efeitos do uso, ou ocorrência do símbolo ou efeitos de outras ocorrências de símbolos no UDI.

As Entradas do LAL descrevem símbolos do UDI e dependendo do símbolo que elas descrevem elas podem ser classificadas como sujeito , verbo , objeto e estado(predicativo). Ao se descrever entradas do LAL deve-se seguir aos princípios de “vocabulário mínimo”(minimização do uso de símbolos externos ao UDI) e “circularidade” (maximização do uso de símbolos do UDI para descrever símbolos do UDI).

A forma natural de representação do LAL é o **hipertexto**. As entradas são os nós do hipertexto e os símbolos que aparecem nas descrições de símbolos são os “elos” do hipertexto.

Extraímos a figura 1 de Cysneiros (2001) que apresenta uma tela da ferramenta OORNF (Neto, 2000) para cadastro de uma entrada do LAL para um sistema de controle automático de luzes. “As palavras sublinhadas são elos para outros símbolos do LAL” (Cysneiros, 2001).



A Figura 1: Exemplo de Entrada do LAL de um sistema de controle de luzes utilizando a ferramenta OORNF.

Fonte: Cysneiros (2001)

Abaixo temos a figura 2 mostrando o modelo SADT do processo de construção do LAL. O processo é formado por três atividades : identificar símbolos, classificar símbolos, descrever símbolos .

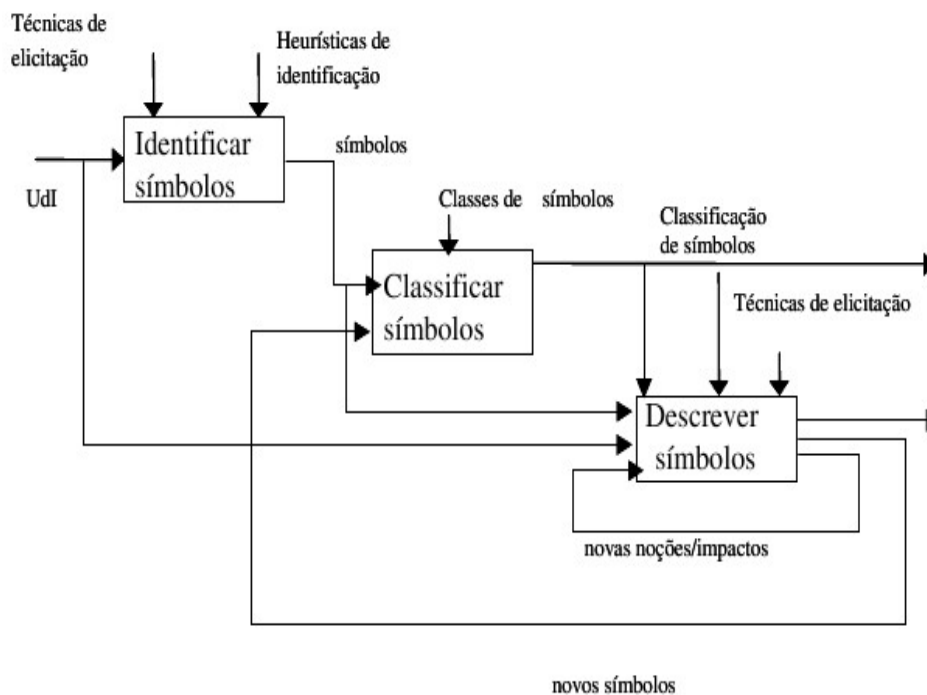


Figura 2: Modelo SADT do processo de construção do LAL do UDI.
Fonte: Cysneiros (2001)

2.4 O RNF-Framework

No NRF-Framework os requisitos não funcionais são tratados como metas possivelmente conflitantes, devendo ser identificados em sua forma mais geral e refinados até que se chegue a um conjunto de requisitos que satisfaçam ao requisito geral Chichinelli e Cazarini (2001).

Durante o processo de projeto, metas são organizadas em uma estrutura de grafos, no espírito de árvores E/OU usados na resolução de problemas. No entanto, essas metas raramente podem ser satisfeitas num sentido bem definido. Decisões de projetos podem influenciar de maneira positiva ou negativa em uma determinada meta. A avaliação dessas metas determina o grau que um conjunto de requisitos não funcionais está sendo apoiado por um projeto particular.

O framework consiste de cinco componentes principais, os quais não estão presentes no SQFD:

- Um conjunto de metas para representar requisitos não funcionais, decisões de projeto e argumentos a favor ou contra as outras metas;
- Um conjunto de tipos de ligações, para referir-se a metas ou relacionamentos de metas (ligações) com outras metas;
- Um conjunto de métodos genéricos para refinar metas em outras metas;

- Uma coleção de regras de correlação para inferir interações potenciais (positiva ou negativa) entre metas ;
- Procedimento de avaliação, o qual determina o grau, através de uma legenda, que determinado requisito não funcional esta sendo conduzido por um conjunto de decisões de projeto;

Para ilustrar o uso do NRF-Framework (ou Framework RNF) retiramos a figura abaixo de Chung (2000):

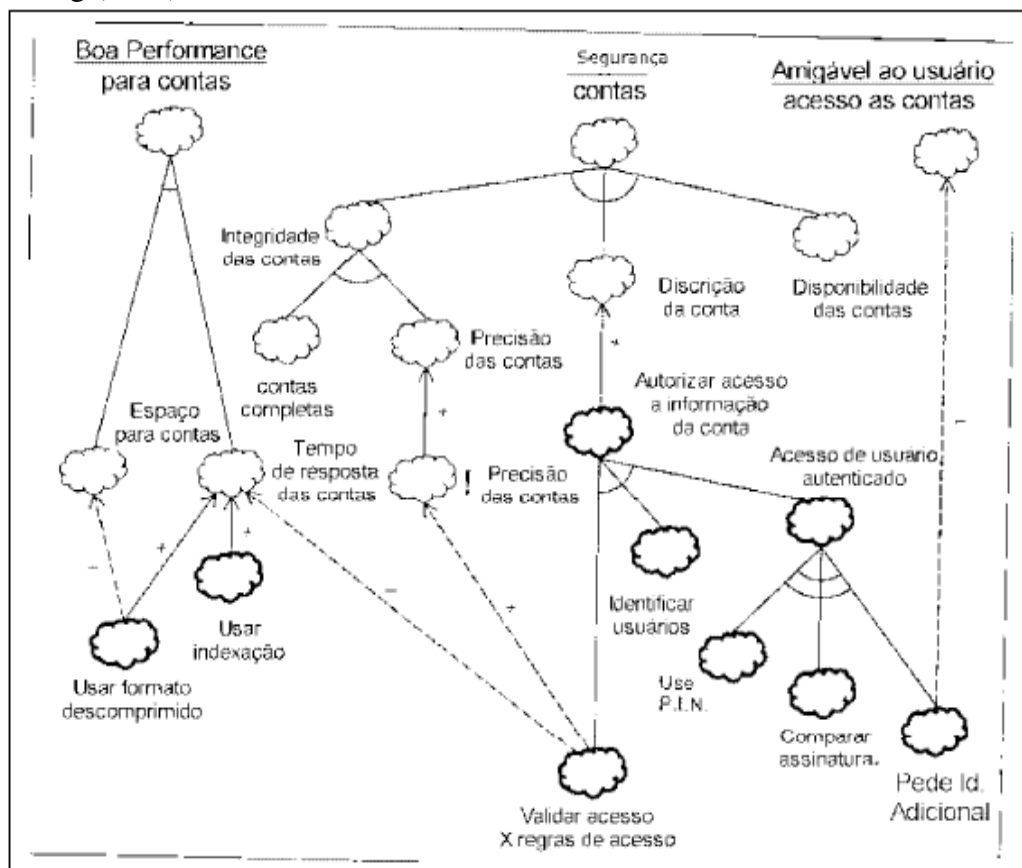


Figura 3: Grafo de RNFs decomposto até as Operacionalizações.

Fonte: Cysneiros (2001)

Nesta figura as nuvens representam metas, que podem ser classificadas em Metas RNF e Metas Satisfatórias ou Operacionalizações (que são representadas por nuvens mais espessas, que são decisões de projeto com o objetivo de satisfazer as metas RNF) (Chichinelli e Cazarini, 2001). Os arcos representam ligações entre metas. Temos dois tipos de ligações a E (que são representadas por um único arco) e as OU (que são as de dois arcos). As ligações E representam que a meta pai é satisfeita, se todas as metas filho forem satisfeitas. Já a ligação OU, implica que a meta pai é satisfeita, se qualquer uma das suas metas filho é satisfeita. Existem as sinergias positivas e negativas. A positiva, que é representada pela seta pontilhada mais o sinal de (+), implica que a operacionalização tem um impacto positivo na Meta RNF. A Negativa, que é representada pela seta pontilhada mais o sinal (-), demonstra que a operacionalização tem um impacto negativo na meta RNF. Estes impactos negativos evidenciam a necessidade de negociação para priorização das metas RNF.

Removemos de Cysneiros (2001) a mesma figura após a análise e negociação para satisfação das metas:

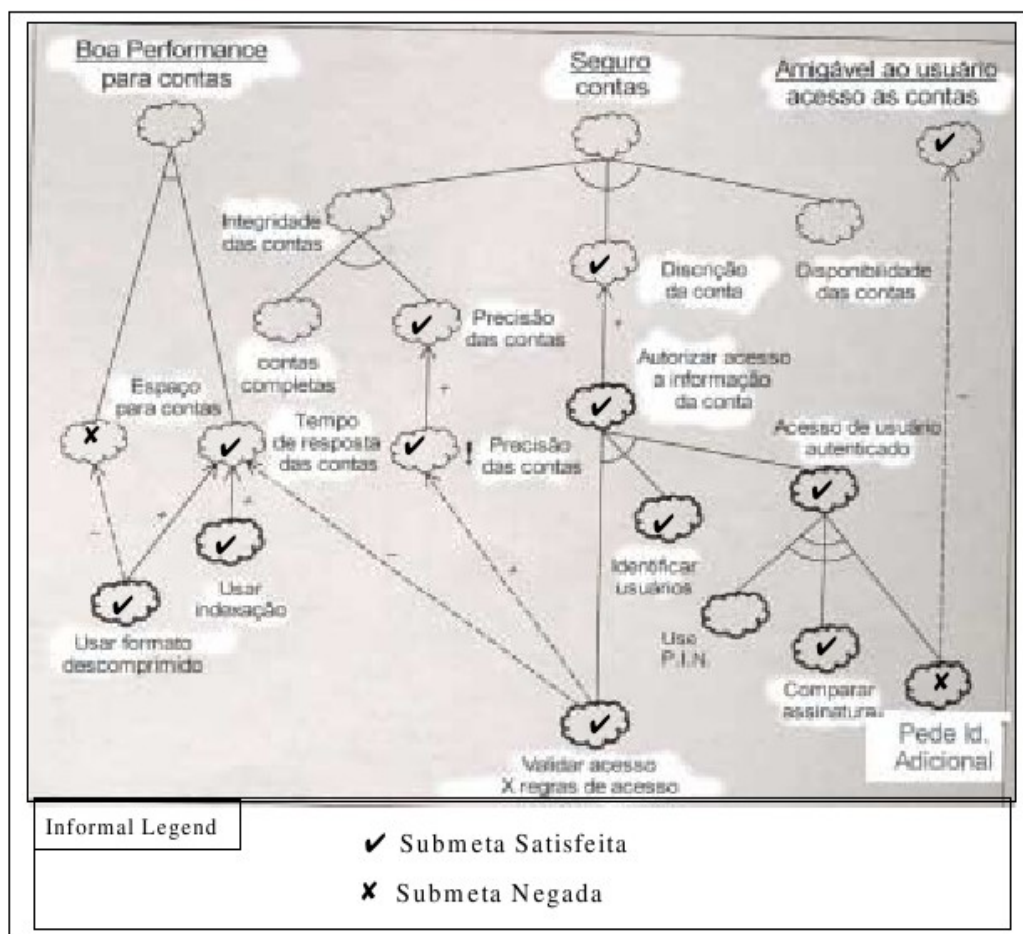


Figura 4: Grafo de RNfs com interdependências analisadas.
Fonte: Cysneiros (2001)

A figura mostra quais metas devem ser satisfeitas e quais delas devem ser negadas.

2.4.1 Geração de Grafos de RNfs

Cysneiros (2001) propoz a utilização do grafo de RNfs proposto por Chung (Chung 1993) e (Chung 2000) com algumas alterações definidas para auxiliar a rastreabilidade de origem e a integração com o modelo funcional.

As alterações propostas para o grafo de RNfs visam a introduzir algum mecanismo de rastreabilidade do RNf à sua origem e a facilitar a integração deste à visão funcional.

Alteração para Facilitar a Rastreabilidade Reversa

Para auxiliar a rastreabilidade Reversa, Cysneiros passou a *representar acima do grafo de RNFs qual o ator do UdI* de onde se originou o conhecimento do domínio que levou-nos a estes RNFs e operacionalizações. Abaixo um figura extraída de (Cysneiros, 2001) demonstrando as origens Administrador do Prédio e CIPA para os RNFs Custos Operacionais e Seguro, respectivamente.

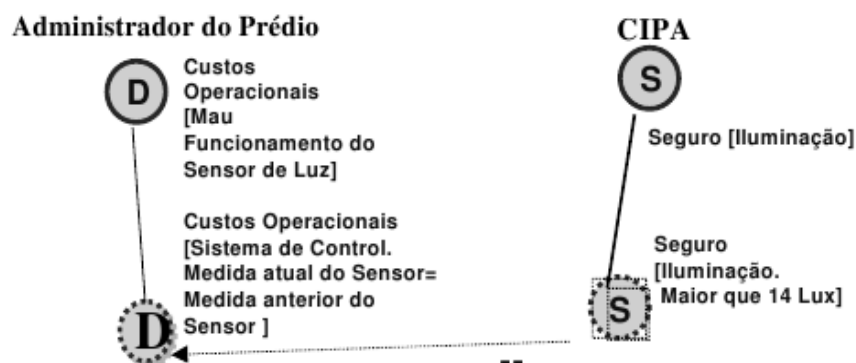


Figura 5: Exemplo de Grafo de RNFs com Identificador de Origem do RNF
Fonte: Cysneiros (2001).

Alterações para Facilitar a Integração das Visões Funcional e Não - Funcional

Para facilitar a integração entre as duas visões Cysneiros propôs uma regra, uma classificação para as operacionalizações, representação escrita para as metas e adaptações na representação gráfica das operacionalizações.

Quanto a regra criada, esta diz que “*Ambas as visões deveriam ser construídas tendo o LAL como âncora*”. Cysneiros faz algumas ressalvas quanto a essa regra que são exibidas abaixo:

No caso do Grafo de RNFs, significa dizer que todo tópico de um RNF será, obrigatoriamente um símbolo do LAL. Quando estivermos refinando as metas em submetas esta restrição não mais se aplica. Contudo, sempre que um símbolo do LAL puder ser usado isto deverá ser feito já que isso resultará numa diferente abordagem a ser utilizada durante a integração das visões (Cysneiros,2001).

Quanto a classificação para as operacionalizações (Uma operacionalização corresponde a ações ou atributos que claramente identifiquem o que é necessário para satisfazer a meta principal), estas são classificadas em “dinâmicas” e “estáticas”. As operacionalizações dinâmicas são representadas por um círculo pontilhado e podem aparecer em qualquer nível do grafo e podem, ainda, serem decompostas em estática. Já as estáticas somente aparecerão no nível folha. A baixo a ilustração e comentário tirados de (Cysneiros, 2001):

A Figura 6 mostra um exemplo onde aparecem ambas as operacionalizações. Nesta figura podemos ver duas operacionalizações estáticas que refinam a submeta de

custos operacionais relacionada a sensores de movimento. Uma delas determina que as luzes devem ser apagadas nas salas depois de T3 minutos vazia e a outra determina que as luzes devem ser apagadas no corredor depois de T2 minutos vazio. Observamos também que na decomposição de custos operacionais relacionados com os sensores de luz, teremos uma operacionalização dinâmica enfatizando a necessidade do sistema usar a luz externa, captada pelos sensores de luz, para manter o ambiente iluminado. Isto nos leva à operacionalização estática que mostra que o esquema de luzes deve ser armazenado também sob forma de quantidade de iluminação, e não apenas na forma de percentual de atenuação das luzes, ou seja, denota a necessidade de um atributo que armazene esta informação.

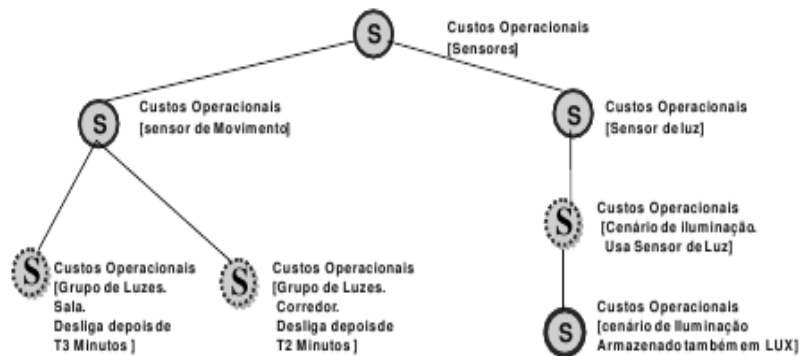


Figura 6: Exemplo de Operacionalizações Estáticas e Dinâmicas

Fonte: Cysneiros(2001).

Cysneiros comenta alteração na notação de grafos

A notação do grafo por nós utilizada é bem próxima da proposta por Chung [Chung 93] e baseia-se na representação do RNF seguido das metas/submetas entre colchetes, ver seção 2.7, onde as decomposições estão separadas por ponto, podendo esta submetas serem omitidas. Da mesma forma que proposto por Chung e explicado na seção 2.7, podemos também decompor metas sob ótica de tipo de RNF (Seguro em integridade e confiabilidade) ou do tópico ao qual ela se aplica (Sensores em sensor de movimento e sensor de luz).

A seção 2.7 da tese de Cysneiros trata sobre conceitos básicos relativos aos grafos de RNF e a disponibilizamos no anexo 1 desta monografia.

Abaixo fizemos um quadro resumitivo sobre a classificação das operacionalizações proposta por Cysneiros:

Classificação de Operacionalização	Associação		Exemplo	Observações	Representação grafica
	necessidade	Diagrama de classes			
estática	a necessidade de um dado específico	atributos	iluminação deve ser armazenada em lux	sempre no nível folha	círculo com borda mais espessa que a normal

	o				
estática	nem dados /n em ação	n/a	num sistema de informações laboratorial, poderemos encontrar um RNF Usabilidade aplicado a Listas cuja operacionalização estabelece a necessidade de a interface de requisição ser a mesma para os diversos tipos de mapas.		
dinâmica	ações a serem realizadas	métodos	desligar a luz depois de T2 minutos	podem aparecer em qualquer nível,inclusive sendo decomposta em uma estática.	círculo pontilhado

Quanto a *representação escrita para as metas*, a Figura 5. 3 ilustra as diversas formas de representação que podemos utilizar. Cysneiros comenta que na figura podemos ver que o **RNF Custos Operacionais** já decomposto para *Sensor de Movimento* é representado sem a sua meta anterior (*Sensores*), sendo representada como Custos Operacionais [Sensor de Movimento]. Existe uma outra forma possível de representarmos esse mesmo nó do grafo que é Custos Operacionais [Sensores. Sensor de Movimento].

O quadro abaixo mostra a diferenciação entre RNF e Meta, bem como a decomposição de metas e as opções para a representação das metas.

	decomposição	Representação 1 (completa)	Representação 2 (simplificada)
		Custos Operacionais [sensores]	igual a primeira
Custos Operacionais	Sensor de Movimento	Custos Operacionais [Sensor de Movimento]	Custos Operacional [Sensor de Movimento]
	Sensor de luz	Custos Operacionais [sensores. Sensor de luz]	Custos Operacionais [Sensor de luz]

Pode-se escolher qualquer uma dentre as opções de representação das metas, cada uma com suas vantagens e desvantagens.

De uma maneira geral a inclusão das metas/submetas anteriores pode e deve ser evitada para efeito de simplicidade da representação do grafo. Entretanto, muitas vezes a representação das mesmas pode exercer um

papel de ressaltar ao que se aplica a submeta sendo representada, o que por vezes auxilia na confecção do grafo (Cysneiros,2001).

Construção do Grafo de RNFs

Um outra contribuição de (Cysneiros, 2001) foi o desenvolvimento de heurísticas para auxiliar na construção sistemática do grafo de RNFs que são apresentadas a seguir:

- Percorrer os símbolos do LAL já com RNFs representados, verificando quais possuem RNFs;
- Para cada símbolo que possuir um RNF faça:
 - Definir a raiz do RNF utilizando o RNF seguido do símbolo do LAL como tópico.
 - Decompor esta meta em submetas seja pela ótica de decomposição em tipos de RNFs (primários em secundários) ou em tópico (sensor em tipos de sensores). Representar o resultado obtido no grafo.
 - Continuar a decomposição até que se possa visualizar o que é necessário para satisfazer este RNF, ou seja, suas operacionalizações. Representar as operacionalizações no grafo de RNF classificando-as em estáticas ou dinâmicas conforme o caso.
 - Nas decomposições anteriores utilizar o conhecimento próprio do domínio e sempre que possível envolver o cliente na definição destas decomposições, avaliando em conjunto com este se as decomposições estão corretas e são suficientes.
 - Verificar no LAL se o RNF não têm consequências (operacionalizações) já definidas seja neste mesmo símbolo, seja em outros símbolos através do uso de opção da ferramenta OORNF de Navegar RNF -> Consequências (ver capítulo 4).
 - Confrontar operacionalizações existentes no LAL com as que foram achadas via passos 2.2 a 2.4. As operacionalizações presentes no LAL que não estiverem no grafo devem ser representadas no grafo, o que eventualmente pode demandar novas decomposições. As operacionalizações existentes no grafo que não se encontrem no LAL devem ser incluídas no LAL. De uma maneira geral, operacionalizações estáticas se traduzem em noções de símbolos e dinâmicas em impactos.
 - No caso de dificuldades em se decompor o RNF (passos 2.2 a 2.4) a identificação de possíveis operacionalizações já descritas no LAL (passo 2.5) pode servir como uma ajuda num processo que alterne abordagens bottom-up e top-down de decomposição deste RNF.

A construção do grafo pode dar-se aos poucos, ou seja, de maneira evolutiva fazendo-se grafos separadamente e depois fazendo-se a convergência posteriormente, se for o caso.

A construção do grafo de RNFs é feita caminhando-se por todas as entradas definidas no LAL que já contenha os RNFs representados. Todo símbolo que possuir

RNFs irá gerar um grafo de RNF, sendo que este RNF junto com o símbolo formarão a raiz de um grafo. Em um momento futuro dois ou mais grafos poderão ser agregados em um só por ter o engenheiro de software verificado que estes grafos são decomposições de um grafo maior. Um exemplo disto pode ser visto na Figura 5.3, onde originalmente tínhamos dois grafos distintos, um para o sensor de movimento e outro para o sensor de luz. Posteriormente avaliamos que na verdade ambos seriam a decomposição do RNF Custo Operacional aplicado a sensores.(Cysneiros, 2001).

Na decomposição por tipos, como a finalidade de nos orientar, devemos utilizar algo que sirva como checklist. Em sua proposta, Cysneiros, (2001) utilizou a base de conhecimento provida pela ferramenta OORNF, apresentada por Neto (2000) e adaptada em Cysneiros, (2001). A ferramenta OORNF mostra quais são os RNFs secundários que decompõe um dado RNF primário.

Extraímos de Cysneiros (2001) o seguinte exemplo:

Tomando como exemplo o RNF Custo Operacional relacionado a sensor de movimento, representado por Custo Operacional [Sensor de Movimento], poderíamos pensar em decompor esta meta nas submetas de sala e corredor, ou seja, teremos que ter operacionalizações diferentes dependendo da localização do sensor de movimento. É necessário ressaltar que, neste caso, as decomposições também são símbolos do LAL, mas isto não é mandatório.

Uma vez decomposta em sala e corredor, temos de verificar se alguma outra decomposição é necessária. Notamos então que temos ainda que especificar como a necessidade de satisfazer a baixos custos operacionais pode ser satisfeita quando observada da ótica do uso de sensores de movimento em salas e corredores. Verificamos, junto ao cliente ou por conhecimento do domínio, que para tal é necessário existirem as ações de desligar as luzes das salas após T3 minutos destas estarem vazias e desligar as luzes do corredor após T2 minutos deste estar vazio. A representação deste processo pode ser vista na Figura 5.5.

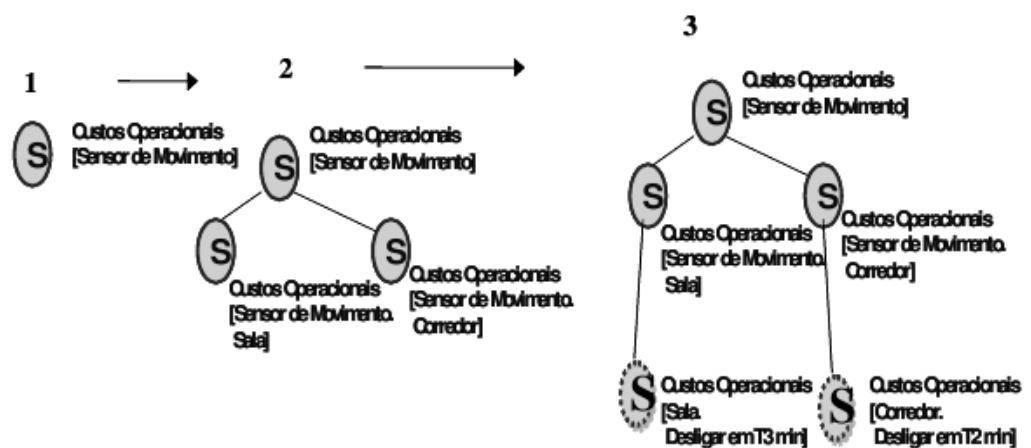


Figura 5.5 – Exemplo da Construção do Grafo de RNFs
Fonte: Cysneiros,(2001)

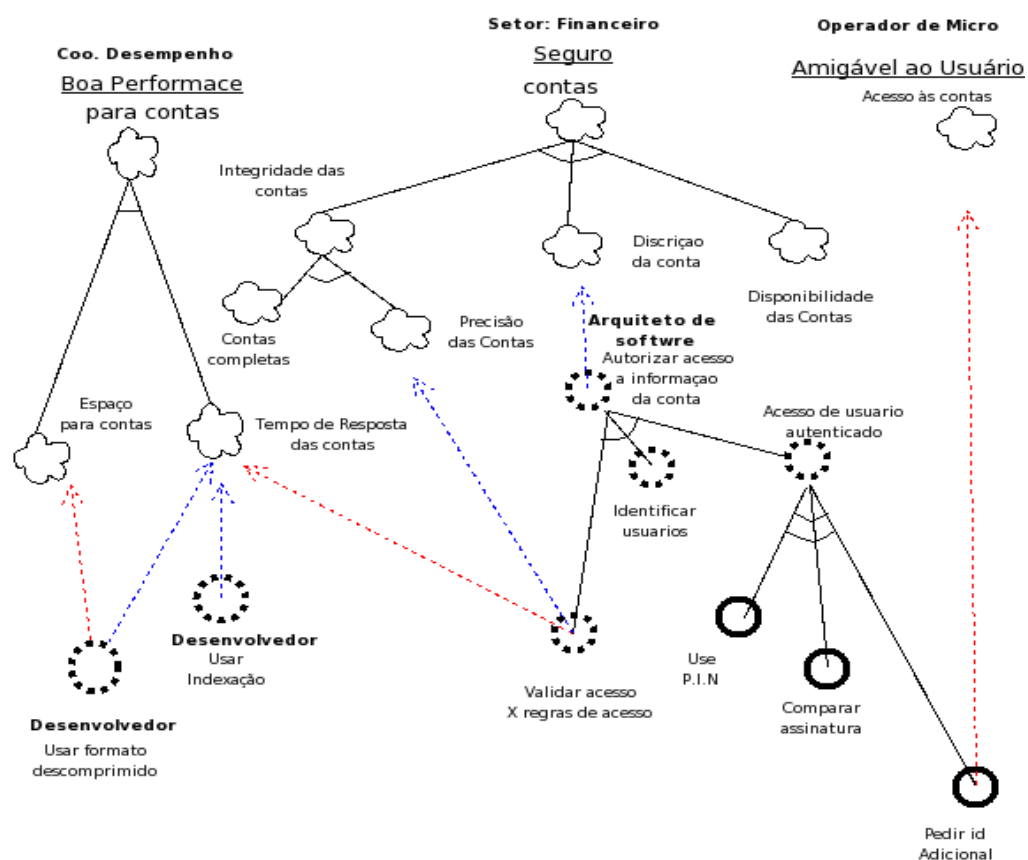
Por fim, devemos verificar no “LAL” quais as “noções” e “impactos” existentes que são consequência da necessidade de satisfazer o RNF Custo Operacional para o símbolo Sensor de Movimento.

É necessário que o engenheiro de software tenha atenção para verificar ao incluir uma operacionalização se esta é refletida pelo LAL. Se isto não acontecer o mesmo deverá ser atualizado para refletir essa operacionalização.

2.4.2 Nossa Proposta de Alteração na Simbologia do Grafo

Propomos melhorias na forma de representação de grafos de RNF propostos por Chung (2000) e nas propostas de Cysneiros (2001). Algumas alterações foram realizadas para ajudar na integração entre as propostas estudadas. Entretanto, mantivemos as contribuições semânticas de Chung (2000) e de Cysneiros (2001). E a simbologia, proposta aqui é muito similar à desses autores.

A figura a baixo mostra o grafo da figura XXX já com algumas de sugestões:



As metas RNF são expostas no formato de nuvem e as Operacionalizações são representadas por círculos com espessura maior que as nuvens. Assim facilitamos a identificação da separação entre requisitos e soluções.

Acrecentamos uma forma de representar que uma operacioanalização está sendo negociada e o andamento da negociação, através da máquina de estados (Ramires, 2004).

A figura abaixo é uma legenda que informa o significado de cada símbolo utilizado na figura anterior.

Figura xx: Legenda para o grafo da figura xy.

2.5 Analise de Negócio: Integração dos RNFs aos Modelos de Processos de Negócio

Neste tópico serão abordados o conceito de modelo de negócio e modelo de processo de negócio. Também será comentado sobre a importância do uso de modelagem de processos de negócio como atividade integrante da fase inicial da Engenharia de Requisitos e como ela ajuda à “comunicação” entre engenheiros de software e solicitantes de software, e consequentemente, contribui para uma boa negociação da qualidade do software e do processo inicial de criação do software. Também será apresentado o resumo (conceito, contribuições e o processo) da abordagem BPMNRNF.

2.5.1 Modelos de negócio

Laís Xavier (Xavier, 2009) cita Davenport e Stoddard, Eriksson e Penker, Berio e Vernadat, Paim e Bittencourt ao apresentar o conceito modelo de negócio. Segundo Xavier (2009) modelos de negócio são “conjuntos de modelos que representam uma abstração da realidade de uma organização sob o ponto de vista do negócio e representam o ponto de vista do negócio, e representam o ambiente no qual a organização está inserida” (Xavier, 2009, p 7).

Xavier também fala sobre alguns benefícios, em relação à tecnologia da informação, do uso dos modelos de negócio. Entre as contribuições estão: permitem compreender as interfaces organizacionais e o fluxo de informações através das unidades de negócio; contribuem para evitar a redundância em sistemas; contribuem na integração de base dados; e permitem descrever o processo de desenvolvimento de um produto. Em sua proposta foi importante a aplicação dos modelos de negócio no desenvolvimento de sistemas, pois permitem, segundo Knight, unificar o conhecimento dos envolvidos e entender as necessidades que o sistema tem (2004 apud Xavier, 2009, p. 8). Em particular, Xavier concentra-se nos modelos de processos de negócio que têm sido vistos como uma forma de integrar as linguagens dos profissionais (o que ajuda a resaltar a importância da análise de

negócio como uma das atividades iniciais da Engenharia de Requisitos) que têm visão de negócios e dos profissionais com visão tecnológica (Xavier, 2009; Davenport and Stoddard, 1994; Paim, 2007)

2.5.2 Modelos de Processos de negócio

Em sua dissertação (Xavier, 2009, p. 9) faz uma definição e explora o

conceito de processo constituído por um conjunto de atividades iniciadas em resposta a um evento, ordenadas no tempo e no espaço, com entradas e saídas claramente identificados, gerando produtos, serviços ou informação. Um processo cumpre um objetivo, seja da organização ou da aplicação que será gerada, e deve ser executado levando-se em consideração o escopo em que está inserido e suas particularidades. “ (Davenport, 1993 apud Xavier, 2009, p. 9)

As *atividades, atores, eventos, regras, recursos e artefatos de entrada e saída* são elementos utilizados no modelo de processo de negócio que mais se destacam. Xavier faz referencia a Bittencourt ao explicar que as informações utilizadas para especificação de um sistema de informação vêm dos elementos de negocio:

as atividades se traduzem em funcionalidades; artefatos que são transformados pelas atividades e se transformam em entidades de informação manipuladas pelo sistema; e atores que representam os elementos (cargos, organizações, sistemas equipamentos) que interagem com o sistema (Bittencourt, 2009 apud Xavier, 2009).

2.5.3 A importância do uso de modelagem de processos de negócio para a Engenharia de Requisitos

Percebemos que os modelos de processos de negócio ajudam a viabilizar uma melhor percepção dos fluxos de negócios, permite o alinhamento dos pontos de vistas dos leitores (na medida em que em que os leitores se guiam pelos mesmos modelos), ou melhor permitem que diferentes leitores tenham o mesmo entendimento ao ler os modelos.

Tais modelos são o ponto de comunicação entre profissionais que têm visão de negócio e os profissionais de visão tecnológica. Uma boa comunicação entre tais profissionais é vital para o sucesso das negociações no decorrer do processo de produção do software. Particularmente, no caso de negociação de qualidade do software, na fase de Engenharia de Requisitos a integração de requisitos não funcionais aos modelos de processo de negócio ajuda, de certa forma, a formar uma linguagem em comum entre os profissionais que precisam que haja eficiência e eficácia da comunicação.

2.5.4 A Abordagem BPMNRNF

Em sua tese, intitulada de “Integração de Requisitos Não-Funcionais a Processos de Negócios: Integrando BPMN e RNF, Laís Xavier propõem uma abordagem chamada de BPMNRNF, em que integra os requisitos Não-Funcionais à notação BPMN. Esta abordagem proporciona uma solução à deficiência de expressividade dos RNFs, isto é, define um processo que viabiliza a inserção de operacioanalizações dos requisitos não-funcionais nos diagramas de processos de negócio (Xavier, 2009). Essa integração proporciona uma melhor comunicação entre os stakeholders, principalmente entre os engenheiros e solicitantes do

software. Entre as contribuições da proposta de Xavier (2009), é o fato de que a proposta evidencia a importância do uso de modelagem de processos de negócio como um passo inicial no processo de Engenharia de Requisitos.

2.6 O SQFD

3 Negociação de Requisitos de Software

Neste capítulo procuramos apresentar alguns conceitos básicos para que sirvam como um arcabouço lógico para abordar a questão da negociação. Depois apresentaremos os principais aspectos presentes no processo de negociação. Por fim as considerações finais serão mostradas.

3.1 Aspectos Conceituais

Aqui nesta seção, falaremos a respeito de alguns conceitos fundamentais de negociação. Tais conceitos possibilitarão aos participantes do processo de desenvolvimento de software se situarem e perceberem o processo de negociação e seus aspectos quando estiver ocorrendo.

3.1.1 Conceitos Fundamentais de Negociação

Carvalho, Neto, Andrade e Araújo (2007) citam conceitos de negociação de autores como Fisher, Cohen, Nierenberg, Lax e Sebenius etc os quais destacam pontos diferentes em suas definições. Dentre os pontos destacados nas definições estão: *comunicação, disponibilidade de informação, comportamentos de influência, tempo, intercâmbio de idéias*(com o fim de afetar o comportamento do outro), *interesses compartilhados, interesses antagônicos etc...* Abaixo algumas das citações feitas no trabalho de Carvalho et al , (2007):

- “Negociação é um processo de interação potencialmente oportunista, pelo qual duas ou mais partes, com algum conflito aparente, buscam fazer o melhor acordo através de ações decididas conjuntamente ao invés do que poderá ser feito por outras maneiras”(Lax e Sebenius, 1986) apud Carvalho et al, (2007).
- “Negociação é o uso da informação e do poder com o fim de influenciar o comportamento dentro de uma *rede de tensão*(Cohen, 1980)
- “Cada vez que pessoas trocam idéias com o intuito de modificar suas relações, cada vez que chegam a um acordo, estão negociando. A negociação

depende da comunicação e ocorrem entre pessoas que representam a si ou a grupos organizados” (Nierenberg, 1991).

Depois do claro entendimento do conceito de negociação é importante o entendimento de outros conceitos como: Necessidades e interesses, Objeto e Escopo, Objetivos e Posições, Moeda de Troca, argumentos, pontos de recuo e MACNA.

Segundo Carvalho et al (2007), o termo *necessidade* assume um significado abrangente, podendo englobar os interesses, desejos, paixões e temores das partes.

O objeto de negociação refere-se ao escopo do que está ou não em questão. Pode ser tangível ou não. O objeto pode ser um bem, um serviço, atributos de qualidade e prazos, conjunto de regras, ou pauta de uma reunião. Também pode ser uma ideia, um caminho de ação, um maneira de punir, um critério para dividir lucros etc. O objeto de negociação caracteriza o que está em jogo, o que gera conflitos, ou oportunidades, o que é considerado importante ou que se tem em vista. Já o escopo é o trabalho que deve ser realizado para entregar um produto com características e funções específicas (Guide PMBOK, 2004: 362).

A explicitação e tangibilidade (considerando a viabilidade) do que se deseja em relação ao objeto caracterizam os objetivos e as posições. As necessidades e interesses determinam os objetivos. Objetivos diferentes podem sanar necessidades iguais, por isso o ideal é que se dispunha de objetivos alternativos para que se busque minimizar as ocorrências de situações de impasse. E as posições estão relacionadas com a flexibilidade e fixação das partes em seus objetivos e são suportadas por argumentações.

As argumentações são consequência do raciocínio destinada a provar ou refutar uma proposição (posição) (Carvalho et al, 2007). Tratam-se de instrumentos para o convencimento das partes. O objetivo da argumentação é fazer com que a(s) outra(s) parte(s) modifiquem suas posições (movimentação), fazendo concessões. Os negociadores, nas argumentações, fazem uso de moeda de troca (concretas ou sentimentais) estas possibilitam a saída de situações críticas.

Mas nem sempre é possível chegar a um acordo. Quando não se tem acordo é por que a negociação chegou no ponto de recuo (quando as necessidades não são satisfeitas). Neste caso o negociador precisa estar precavido e ter uma MACNA, melhor alternativa em caso de não acordo, para retirar-se da negociação e não aceitar as imposições por falta de alternativa.

3.2 Conflitos: tipos, causas

Para que haja negociação é necessário a ocorrência de conflito pois “negociar é tratar de interesses antagônicos e comuns em torno do que se dá valor, é resolver conflitos e aproveitar oportunidades que vão surgindo” (Carvalho, 2007). O conflito se torna, de certa forma o ponto central da negociação, ou seja a razão de haver negociação. **Momentos de tensão decorrem da expectativas do envolvimento das pessoas em situações conflituosas (Carvalho, 2007).** Existem vários tipos de conflitos e para resolver os conflitos é necessário identificar suas causas e escolher a melhor forma de intervir.

Segundo Carvalho (2007), os “tipos de conflitos que impactam as relações organizacionais são os: interpessoais, os intergrupais, os intrapessoais e interorganizacionais” e os conflitos objetos de abordagens cotidiana dos líderes de projetos, de gerentes e

administradores são os *interpessoais e intergrupais*. Para este trabalho serão abordados os tipo interpessoais e intergrupais.

Robbins (apud Camacho, 2005) distigue os conflitos entre conflito funcional e conflito não funcional, referido-se à pertinência do conflito para a organização.

As causas para os conflitos são de dois tipos: causas reais e causas emocionais (Carvalho, 2007). Já Robbins (apud Camacho, 2005) diz que as causas podem estar entre: fatores de comunicação, estruturais e pessoais.

3.2.1 Conflitos de Requisitos

Durante o exercer das atividade de levantamento, registro, validação e verificação, citadas acima, os engenheiros de software podem identificar diversos conflitos.

O Swebok (2004) classifica os conflitos (referindo-se especificamente ao processo de desenvolvimento de software) em três tipos: entre dois stakeholders que requerem características incompatíveis entre si; entre requisitos e recursos; ou entre requisitos funcionais e não funcionais.

3.3 Abordagens de Negociação

As abordagens de Negociação são dispostas em duas: teóricas e práticas. As teóricas subdividem-se em prescritiva e descritiva. E as práticas em integrativa e distributiva. Nossa abordagem do tema se apóia em abordagens práticas (Ramires, 2004).

3.4 Aspectos de Relacionais e Substantivos

“Todo negociador tem dois tipos de interesse: na substância e na relação” (Fisher, Ury, Patton apud Carvalho, 2007). É fundamental desenvolver a habilidade de indentificar os planos onde ocorrem os conflitos e de que maneira as abordagens podem ter uma probabilidade de maior êxito.

O plano onde ocorre os conflitos a ser dado maior foco, neste trabalho, é o plano substancial (requisitos de qualidade do software), mas é importante observar que o relacional e o substantivo devem ser considerados ao mesmo tempo quando se quiser que haja efetividade nas negociações. Para esta monografia é dado o foco ao plano substancial (racional) por motivos de escopo de pesquisa e tempo. Percebe-se que, ao melhor tratar a substância, o relacional também é beneficiado.

A seguir falaremos de processo de negociação, os estágios e a abordagem de Ramires (2004), aplicada ao processo de desenvolvimento de software, pois “a manifestação mais intensa dos aspectos relacionais ou substantivos se altera-se no decorrer do processo, por isso compreender a natureza das fases e dos estágios é essencial” (Carvalho et al, 2007).

3.4.1 O processo de Negociação

Em Carvalhal et al (2007), processo de negociações pode ser dividido em fases: Planejamento, Execução e Controle. A fase de Planejamento tem os seguintes estágios: planejamento e preparação. A fase de Execução tem os seguintes estágios: preliminar, abertura, exploração, encerramento. E a fase de Controle possui como estágios: o controle de condições e avaliação. Em negociações simples o processo é intuitivo. Mas em negociações complexas cada fase é detalhada em estágios e cada estágio é trabalhado. No planejamento o negociador pensa na espinha dorsal que sustenta suas ações.

Em Ramires (2004), João Jorge Ramires apresenta um processo para negociação de requisitos que consiste em três fases: Pré- negociação (onde surge o conflito), negociação (onde há troca de informação e solução proposta) e pos-negociação (quando o acordo é firmado). Este processo faz parte de sua proposta (que será abordada mais à frente) para a resolução de conflitos na avaliação de requisitos de software.

3.4.1.1 Sistema de apoio a resolução de conflitos

Ramires (2004), propõem um sistema que interliga grupo de pessoas e suporta técnicas de negociação na resolução de conflitos que surgem na avaliação de requisitos de software.

Em sua proposta é apresentado um sistema (sistema MEG) de suporte a negociação (NSS) e tomada de decisão em grupo (GDSS). Sua abordagem é descrita no contexto da Casa da Qualidade (mas o processo a aplicar nas demais áreas é análogo).

O sistema visa o auxílio à resolução de conflitos de *corr* entre stakeholders e a obtenção de um valor de consenso. O sistema também propõem um valor baseado em voto maioritário e comportamento distributivo. A Negociação no Modelo SQFD, Máquina de Estados e Processo de Negociação formam a base da solução apresentada por Ramires (Ramires, 2004).

Negociação no modelo SQFD

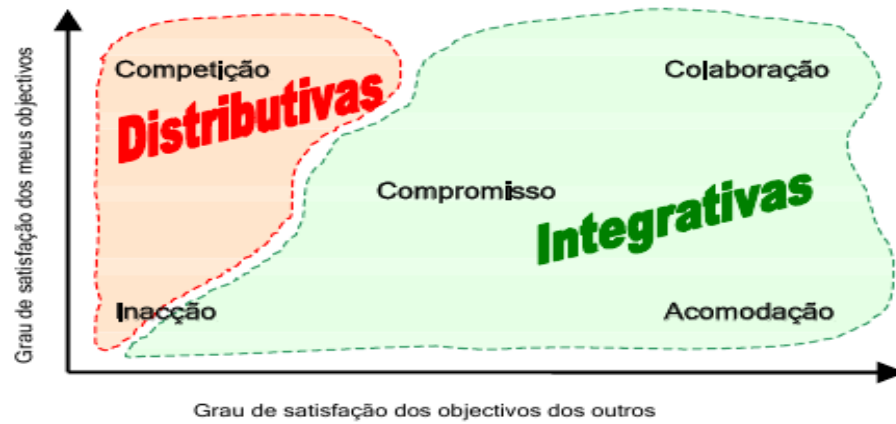
Ramires (2004) apresenta um modelo de negociação baseado no SQFD. Este modelo usa matrizes nas quais se comparam os requisitos com as soluções na forma de correlações (*corr*). Para as correlações são possíveis os valores *nenhum*, *fraco*, *médio* e *forte*, o que corresponde, respectivamente a 0,1,3 e 9.

Stakeholders diferentes podem atribuir valores diferentes para *corr*, o que pode causar conflitos de interesses. O sistema MEG auxilia a resolução de conflitos de *corr* e obtém um valor para *corr* por consenso. O MEG também propõe um valor baseado em voto maioritário.

O sistema assume que o processo se desenrola numa CÉLULA de swHoQ em que a CÉLULA pode estar em um estado de equilíbrio (E) ou de negociação (T)(serão abordados os estados da célula mais detalhadamente no tópico Máquina de Estados). O swHoQ é a casa de qualidade do software e é representada por $swHoQ = \{ \{r, c, corr\} \}$. O “r” corresponde ao requisito e o “c” à característica que o software terá ao ser implementado.

O sistema também suporta e controla as atitudes dos stakeholders que podem ser de acomodação, competição, inação, colaboração e compromisso. Estas atitudes podem ser divididas em atitudes distributivas e integrativas. Dentre as distributivas: competição e inação

distributiva. Dentre as integrativas estão: colaboração, acomodação, compromisso e inação integrativa. Veja a figura:



A partir da análise da figura verifica-se que a o sistema pode permitir o controle do grau de satisfação dos stakeholders, comparando o número de atitudes distributivas com o número de atitudes integrativas.

Mas o problema com essas matrizes é que não passam muita informação, o que faz com que acabe servido somente para a votação e priorização de soluções e interesses. Uma possibilidade é substituir as matrizes por grafos. Sugerimos neste trabalho o uso do RNF-Framework (a estratégia será apresentado em capítulo posterior).

Maquina de Estados

Ramires propõem, para o acompanhamento do processo de negociação, uma máquina de estados. Entre os estados possíveis da Célula estão as subdivisões: para equilíbrio (E), temos E(a) e E(b); para negociação (T), temos T(n), T(o), T(p), T(x) e T(z). Formalmente a célula é representada por

$$\text{CÉLULA} = \{ \{E | T\}, \{ \text{corr}, \text{SCÉLULA} \} \}$$

A **figura** abaixo representa a passagem de estado. Em seguida a descrição retirada de Ramires(2004):

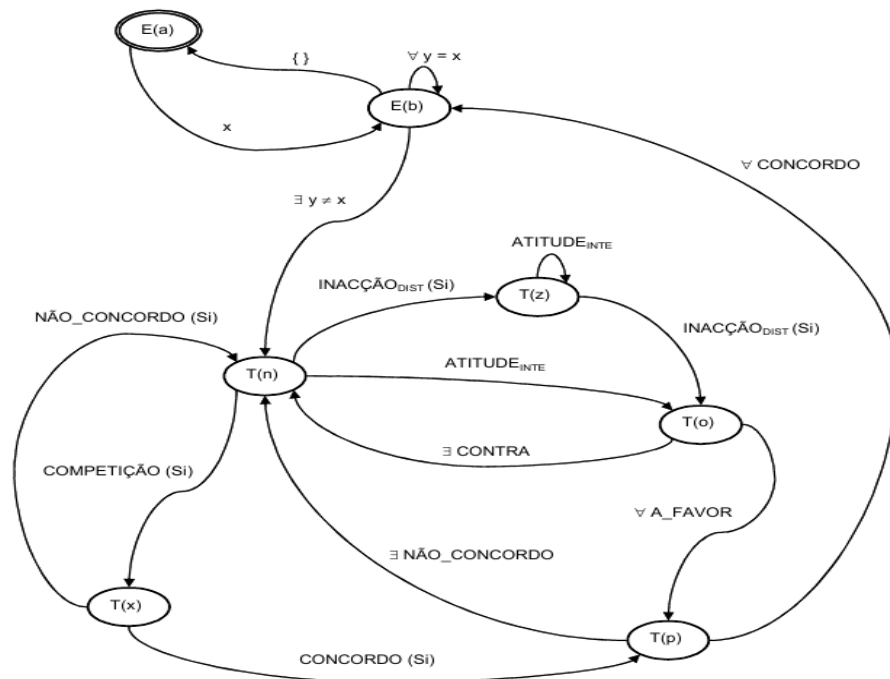


Figura 5.3 Máquina de Estados

Inicialmente, uma célula encontra-se vazia – estado E(a)¹. Por inserção de um valor x na célula, passa-se a um estado de célula preenchida E(b)². A inserção de novos valores y

¹ Esta situação pode ser descrita formalmente como $\text{CÉLULA} = \{ E(a), \{ \} \}$

² Ou seja, $E(a) \rightarrow x \rightarrow E(b)$

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 $E(b)$ ³ 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 $T(n)$ ⁴. Senão, mantém-se o estado $E(b)$ ⁵.

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

Numa atitude integrativa $ATITUDEINTE$ 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 $T(n)$.

No entanto, num estado de conflito $T(n)$, os stakeholders podem tomar uma atitude distributiva $ATITUDEDIST$. Uma atitude $COMPETI\tilde{C}A\tilde{O}$ explícita por um stakeholder S_i ($COMPETI\tilde{C}A\tilde{O}(S_i)$) 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 ($N\tilde{A}O_CONCORDO(S_i)$) retorna-se ao estado $T(n)$. O autor S_i de $COMPETI\tilde{C}A\tilde{O}(S_i)$ não necessita de confirmar a sua opção no estado $T(p)$.

Por outro lado, se um stakeholder S_i tiver uma atitude $INACCAODIST$ ($INACCAODIST(S_i)$) 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 $INACCAODIST$ 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 $INACCAODIST$ ($INACCAODIST(S_i)$) passa-se ao estado $T(o)$ onde se verifica se existe um ponto de intersecção entre os intervalos de valores eventualmente criados por $ATITUDEINTE$ em $T(z)$. (Ramires, 2004)

Processo de Negociação de Requisitos

O processo apresentado por Ramires (2004), é composto de três fases: Pré-negociação, Negociação e Pós-Negociação.

A fase de pré-negociação apresenta o problema, ou seja, prepara os pontos a negociar na fase de negociação. É nessa fase que ocorre a *descrição do problema* que 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 $T(n)$.

Na fase seguinte, fase de negociação, os stakeholders inserem seus respectivos valores e argumentos (troca de informação). É nessa fase que ocorre a *Troca de informação* que corresponde ao estado $T(n)$: pode-se “inserir mais do que um valor”, o que permite o “alargamento de valores possíveis” com o fim de encontrar um ponto de intersecção entre os

3 $C\acute{E}LULA = \{E(b), \{\{corr(x), S_i\} \mid \{\{corr(x), S_i\}, \{corr(y), S_j\}, x=y\}\}\}$

4 $E(b) \rightarrow x \neq y \rightarrow T(n)$

5 $E(b) \rightarrow x=y \rightarrow E(b)$

interesses. Também é nessa fase que “surge uma solução”, quando surge, é “necessário obter o consenso dos stakeholders envolvidos na discussão (caso contrário continua a troca de valores / argumentos)” (Ramires, 2004:94), nesse momento o estado da Célula é T(o). Segundo Ramires, a etapa de troca de informação representa atitudes implícitas de COMPROMISSO, COLABORAÇÃO e ACOMODAÇÃO, à medida que 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 faz com que o ao estado da CÉLULA vá para 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 INAÇÃO_{DIST} impossibilita a passagem à fase de pós-negociação, ou seja, não permite que se chegue a consenso.

A figura abaixo exhibe o processo proposto por Ramires :

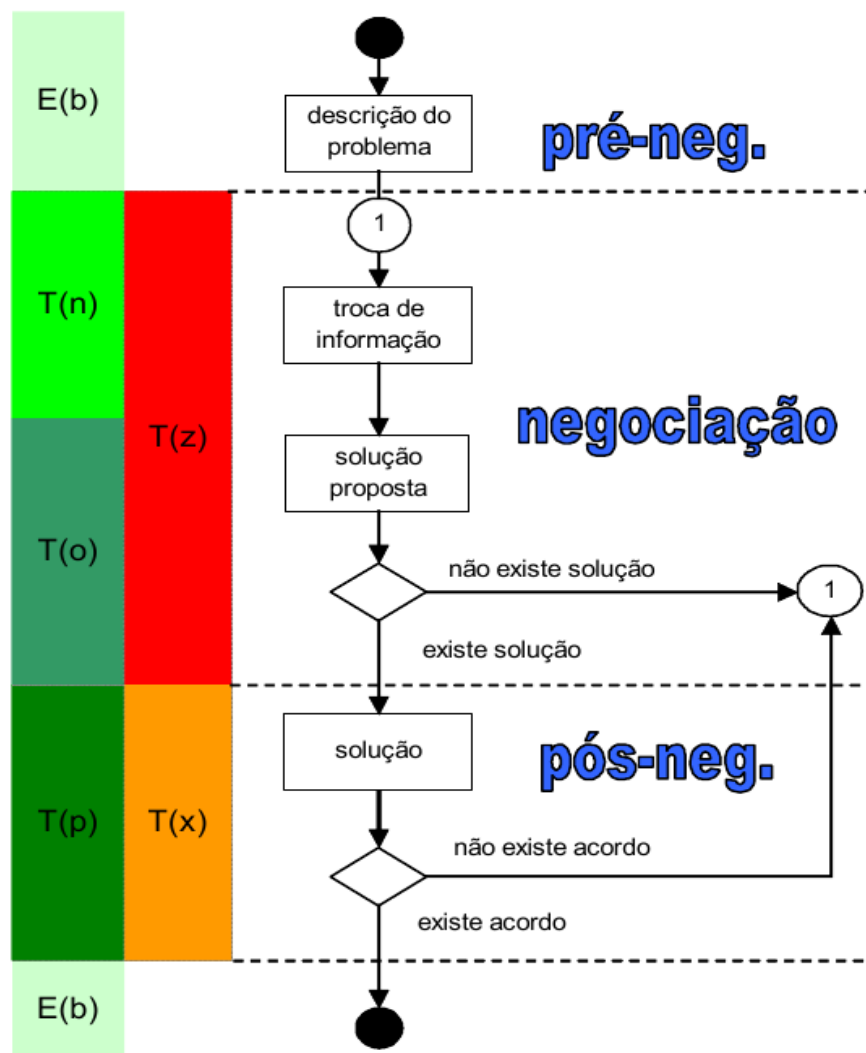


Figura 5.4 - Processo de negociação

A última fase do processo é a fase de pós-negociação, nesta fase tenta-se chegar a um consenso quanto a “solução proposta” na fase de negociação. Se houver consenso o valor encontrado passa a ser a “solução”, então passa-se ao estado de equilíbrio E(b) até ser introduzido um novo valor que entre em conflito com o valor acordado, se isso ocorrer o processo é reiniciado, então volta-se ao estado T(n).

Conforme Ramires, 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 Si com atitude COMPETITIVA em T(n), sendo o valor proposto o correspondente ao valor de Si.”

Estratégias e táticas nas Negociações

- Normalmente , não se altera a estratégia (somente quando as circunstâncias mudam);
- Quando o número de atores e interesses é grande, um mapa de partes se faz adequado;

O modelo de Partes possui quatro dimensões:

- vertical interna;
- interna;
- horizontal e;
- vertical externa.

A conexão entre o objeto de negociação, as partes envolvidas e os interesses em jogo em cada dimensão requer um planejamento estruturado. Ele se inicia nos aspectos mais gerais e se dirige para os mais específicos.

Planejamento e Organização

- análise ambiental;
- análise quantitativa;
- análise qualitativa;
- planejamento tático;
- Simulação (negociações complexas);

Posicionamento estratégico e passos táticos:

- uso do poder;
- fuga;
- amaciamento;
- barganha;
- integração;

Desequilíbrio, comportamento ético e comunicação

- O uso da informação , do tempo e do poder;
 “O sucesso provém não apenas de ter boas cartas, mas da análise de toda situação, para que cartas possam ser jogadas com habilidade”.(XXX, aaaa).
- Manipulação e comportamento ético nas negociações:
 - a médio e longo prazos os manipuladores concientes tendem a perder a credibilidade;

Intercâmbio de informação

- negociadores habilidosos passam 38,5% do seu tempo obtendo e esclarecendo informações;
- quando a abordagem competitiva é predominante, o tratamento da informação afetará o desenvolvimento do processo e seus resultados (pela omissão, retardamento ou distorção);
- quando a abordagem integrativa é predominante , o tratamento da informação (pela presteza, objetividade, e clareza) contribuirá para que a maximização dos resultados e dos relacionamentos possa ser alcançada;

Maximização de resultados e relacionamentos

Quando há compartilhamento de dados e informações sobre suas respectivas necessidade e interesses , as alternativas , a possibilidade de encontrar um ponto em de equilíbrio ótimo aumenta.

A busca do resultado ótimo

- jogos “ganha-ganha”
- atingir a satisfação de suas necessidades, sem prejudicar os resultados de outras partes
 - abordagens cooperativas

ASPECTOS SUSBSTANTIVOS

São associados ao conteúdo e são determinantes para qualificar o objeto e os resultados concretos que se espera obter ao final de uma negociação. Podem ser classificados em dois grandes grupos: **tangíveis e intangíveis**.

Entre os principais aspectos tangíveis, podemos destacar aqueles ligados ao **escopo, aos produtos e serviços** que estão em jogo. Já intangíveis estão ligados aos **níveis de satisfação** das partes com a natureza do processo de negociação e sua qualidade, **pauta e agenda, estratégia e processo decisório**, além dos **cronogramas** e das **metas intermediárias** a serem atingidas.

Negociação no plano substantivo

É fundamental conhecer a maior quantidade de informações associadas à satisfação de suas necessidades.

Objetos de negociação:

- características técnicas e qualidade esperada de produtos e serviços
- quantidades,
- garantias
- prazos,
- assistência técnica,
- apoio operacional
- preços
- tributos e taxas
- tarifas
- acesso a mercados e formas contratuais
- etc.

Outros fatores influenciam na quantificação e qualificação dos resultados :

- conhecimento do setor onde se insere o objeto;
- atuais e principais fornecedores e clientes;
- equilíbrio poder existente (no mercado) e contexto específico.

São objetos preliminares de negociação:

- a pauta;
- a agenda, cronogramas ou fases;
- local onde a interação se dará e;
- a forma a ser adotada para o processo negociador avançar e produzir seus efeitos.

(Esses objetos) Podem ocorrer de forma sequencial ou agrupados.

Visão geral das áreas de negociação em gerenciamento de projetos

1. Integração	2. Escopo	3. Tempo
4. Custos humanos	5. Qualidade	6. Recursos
7. Comunicações	8. Riscos	9. Aquisições

Fonte: adaptado de Guide PMBOK (2004).

As etapas na negociação podem ser divididas em :

Fases	Estágios
Planejamento	Planejamento Preparação
Execução	Preliminar Abertura Exploração Encerramento
Controle condições	Controle das Avaliação

“Seja qual for o objeto de negociação, as fases não se alteram”. (Carvalho et al, 2007).

A importância do escopo

Planejando a negociação do escopo

Critérios objetivos na negociação do escopo

Cuidados ao negociar o escopo em projetos complexos

A construção negociada do escopo

A fórmula de sucesso em uma negociação

$$S = T(\text{tempo/oportunidade}) * I(\text{informação}) * P(\text{poder})$$

Melhor alternativa em caso de não-acordo (Macna)

A busca por melhores alternativas que atendam a qualidade, prazos, e preços favorecem o fortalecimento da MACNA.

Negociação de qualidade

Negociação de prazos

Negociação de preços

Negociação de aquisições – a compra e venda consultiva

Dicas para a negociação de aspectos substantivos

Outros resultados possíveis

Comentário 1

Todo negociador tem dois tipos de interesses : na *relação* e na *substância* (Carvalho, 2006). Podemos fazer as seguintes inferências : que o uso de grafos de RNF nas negociações ajuda a aumentar a confiança, a comunicação e o interesse de participação das negociações entre os stakeholders e portanto, ajuda o *aspecto relacional* da negociação de requisitos. Por si só, o uso dos grafos de RNF nas negociações reforçam a relação entre os stakeholders e partes⁶ envolvidas nas negociações, porque são fáceis de entender, e são ricos em informações, além de representarem as correlações requisito/ solução (nesse ponto são mais

6 em Carvalho (2006) há uma diferenciação entre stakeholders (“fantasmas”) e partes envolvidas diretamente nas negociações.

ricos que as matrizes de correlação requisito/solução do modelo SQFD, pois estas somente expressam as correlações requisitos/solução). Tomados em conjunto com o LAL, UML e GRAFOS RNF, reforçam ainda mais o *aspecto relacional*, pois esses, usados em conjunto (tal como em Cysneiros (2001)) permite a integração entre as visões funcional e não funcional do software e, portanto, facilita a associação de RNFs à visão funcional (que é a visão que o utilizador tem do software). Assim, pode-se tomar decisões de projeto, de forma lúcida, antecipada e negociada, que terão impacto na visão funcional com a possibilidade de se saber como o software irá se comportar e antes que se construam o software. Pode-se então evitar ou diminuir as tensões decorrentes das expectativas de envolvimento das pessoas em situações conflituosas.

Comentário 2:

As abordagens de negociação podem ser natureza teórica ou prática. Kemper e Kemper (apud Carvalho(2006)) classificam as abordagens teóricas em nove categorias mais importantes: estratégicas (estratégias de jogadores); conhecimento do processo de aprendizagem; processo de tomada de decisão; psicológicas e sociológicas; prescritivas (o que deve ser feito); descritivas (características dos negociadores); teoria dos jogos; comunicação; negociação coletiva. No campo prático as abordagens *teórica prescritiva* e *teórica descritiva* são as mais utilizadas. Já no que se refere a abordagens práticas, os negociadores podem adotar diferentes abordagens. Mas as mais importantes são as de natureza integrativa (*abordagem prática integrativa*) e de natureza distributiva (*abordagem prática distributiva*).

Comentário 3:

Os interesses, necessidades, objeto, escopo, objetivos, posições, moedas de troca, valores, argumentos, limites, ponto de recuo e Macna são conceitos que devem ser bem entendidos e praticados no contexto das negociações de requisitos de qualidade para se garantir a lucidez das negociações. O uso de softwares de apoio é fundamental para que se garanta a produtividade dos negociadores.

REFERENCIA:

Carvalho, E.D, Neto, A.A, Andrade, G.M.D., Araújo, J.V.D., **Negociação e Administração de Conflitos**. Série Gerenciamentos de Projetos, Editora FGV, 1º ed; pag 183, Rio de Janeiro, RJ, Brasil, 2006.

4 Estratégia de Engenharia de Requisitos

5 Projeto Arquitetural

6 Análise da Viabilidade

7 Resultados

8 Conclusões

9 Trabalhos Futuros

As atividades de análise concentram-se na identificação, especificação e descrição dos requisitos do sistema de *software*. Os conflitos que surgem ao decorrer do “ciclo de software” são vários. Por exemplo, podem ocorrer conflitos de interesses dos participantes, de requisitos com requisitos, de requisitos com restrições de implementação.

4 ANÁLISE SOBRE NEGOCIAÇÃO DE REQUISITOS NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

4.1 Classificação de conflitos, suas causas e o Nível de Racionalidade do Processo de Negociação de Ramires.

Na proposta de (Ramires, 2004), é dado um suporte a negociação e resolução de *conflito na avaliação de requisitos de software*. No suporte a negociação esta incluso suporte à argumentação com o objetivo de aumentar a racionalidade do processo de negociação. Mas não há preocupação em se evidenciar as causas e nem em se classificar os conflitos. Quais as implicações disto? Em que ponto(s) classificar e saber a causa dos requisitos é importante para a negociação?

Ramires trata de *conflito na avaliação de correlações requisito/solução de software*. Este pode, segundo nossa avaliação, se dar devido a fatores de comunicação, estruturais e pessoais e pode ser classificado como funcional ou não funcional como qualquer outro conflito segundo o conceito e classificação de conflitos de Robbins (apud Camacho, 2004). E pode se enquadrar em um dos três tipos de conflito citado no SWEBOK Guide (SWEBOK, 2004) e/ou esta relacionados a eles. Mas isso, não é citado na proposta de Ramires. Por que?

Na proposta o conflito se emerge quando stakeholders diferentes atribuem valores diferentes a uma mesma correlação requisitos/solução. Esse valores diferentes

deixam implícita, segundo (Ramires,2004), a posição de “a favor” ou “contra”. Estas tornam evidente a existência do conflito. Após o surgimento do conflito é dado início à negociação (Ramires , 2004). Mas o sistema não provê suporte aos stakeholders para fazerem avaliações sobre a pertinência do conflito para a organização, suas origens e classificação e os impactos das possíveis decisões para a organização.

A resposta à questão anterior talvez(por que Ramires não classifica e nem se preocupa com a causa dos conflitos?) seja pelo fato do autor considerar que essas observações e avaliações devem ser feitas em outro momento anterior à negociação. Ou talvez nem tenha pensado no assunto. Ou, ainda, tenha delimitado o escopo do seu trabalho até o certo ponto em que considerar essas observações não fazia parte dos seus objetivos de estudo. De qualquer forma, tais considerações são relevantes, já que trariam inúmeros benefícios e devem ser realizadas.

Observamos que, devido às formas diversas de se considerar os conflitos, as classificações dos conflitos deveriam ser feitas de forma facetada: quanto a pertinência para a organização; quanto a natureza do conflito;

A proposta de Ramires também oferece suporte à argumentação (apartir do SQFD) com o objetivo de aumentar a racionalidade do processo de negociação, como foi dito anteriormente. Contudo, este suporte a argumentação é limitado e prover um certo nível de racionalidade. Mas esse nível de racionalidade não é quantificado ou qualificado. O sistema provê um modo de os stakeholders exporem seus argumentos com o fim de dar suporte às suas posições sobre as questões na negociação. Mas não auxilia os stakeholders a relacionarem o conflito com os objetivos da organização, não satisfaz a necessidade de se saber o quanto a falta de determinada característica no software vai prejudicar a organização nos seus planos para o futuro, ou seja, não aumenta a racionalidade da argumentação. Enfim, satisfaz a necessidade de se expor argumentos mas não satisfaz a necessidade de suporte à argumentação racional(lúcida).

Percebe-se que esforços para aumentar os níveis de Racionalidade da Argumentação permitiriam verificar se determinada posição seria considerável ou não. Mas a complexidade envolvida no processo de software e os limites cognitivos humanos impõem barreiras à racionalização dos processos de negociação. Para driblar tais barreiras sugerimos esforços com a produção de software de apoio a racionalização de argumentação.

Concluimos que dentre as implicações de não se considerar as causas e nem se classificar os conflitos estão o acarretamento em perda de tempo em discussões irrelevantes, irracionais e disfuncionais etc. Fazendo a classificação e investigação das causas dos conflitos poderíamos, por exemplo, saber quando desconsiderar a negociação de alguns conflitos, perceber a lucidez de uma negociação, alinhar as negociações aos objetivos mais gerais das organizações etc.

4.2 Guiando o processo de desenvolvimento e as negociações

Ramires afirma:

“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.” (Ramires, 2004).

Esta citação é um ponto que reforça a ligação da proposta de Ramires com a de Cysneiros.

Couto & Martins reforçam esta afirmação indiretamente quando afirmam que os desenvolvedores podem usar os RNFs para guiar o processo de desenvolvimento. Estes reforçam suas afirmações referenciando trabalhos de Lamsweerde e Chung:

“O NFR-Framework contribui para que os desenvolvedores tratem os requisitos não-funcionais de modo a expressá-los sistematicamente, e usá-los para guiar o processo de desenvolvimento de software [2] [4]”. (Couto & Martins, 2006).

O [2] refere-se a Chung e [4] referi-se Lamsweerde.

A estratégia proposta por Cysneiros é fortemente apoiada no NFR-Framework proposto por Chung e possibilita uma integração das visões funcionais e não-funcionais. De certo grau de rastreabilidade do impacto que as modificações de desenho dos NFR-Framework provocam na visão funcional. Isto pode ser usado durante as negociações de requisitos para se comprovar a relação entre RNF e requisito funcionais e a importância de se negociar RNF durante o processo de Engenharia de Requisitos.

A proposta de Ramires apresenta uma abordagem para a resolução do conflito na avaliação de requisitos do software. Isto é, trata da negociação para resolução de conflitos de interesses entre stakeholders. Os conflitos são representados por numerações que são atribuídas às correlações requisitos/solução. Ramires baseia-se no modelo SQFD.

Na sintaxe e semântica do NFR-Framework são representada as interdependências entre RNF. Entre estas interdependências estão a interdependências (correlações) negativas de RNF com outro RNF. Estas são as que provocam a necessidade de negociação de requisitos.

O conflito representado por essas correlações são de conflito entre requisitos e não entre pontos de vista diferentes de stakeholders. Mas apesar disso, verificamos que conflitos entre RNF têm impactos sobre a visão funcional do software. E que por traz da visão funcional do software podem haver stakeholders que priorizam atributos de qualidade diferentes, evidenciando assim a necessidade de negociação para saber quais atributos beneficiar. Pois uma decisão de desenho feita sem que haja a negociação entre stakeholders afetados poderia beneficiar uns mais que outros antes mesmo que o consenso fosse estabelecido.

Percebemos, então que o sistema proposto por Ramires pode ser usado em conjunto com a proposta de Cysneiros. Este para tratar de como elicitar RNF e integra-los ao modelo conceitual, dando embasamento para a negociação. E aquele para prover suporte à negociação e tomada de decisão em grupo.

Acreditamos que a utilização em conjunto de tais propostas possam elevar o nível da negociação de requisitos ao ponto que os stakeholders possam relacionar artefatos de visão funcional à atributos de qualidade.

4.3 PROCESSO ITERATIVO

O processo de negociação deve ser iterativo, pode possuir, então, sub-negociações, renegociações. Ramires propoem um processo de negociação na sua aborgem.

4.4 O ENTENDIMENTO DO *QUE SE PEDE* E AS CAUSAS DA INSATISFAÇÃO DOS CLIENTES

Ramires(2004) procurou solução para o problema :

“ falta de entendimento entre o que de facto de 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.”(Ramires, 2004)

O enunciado sugere que o software entregue não é de acordo com as expectativas dos stakeholders requerentes, o que é um problema, e que isso é causado pelo fato de que os stakeholders requerentes do software pedem uma coisa mas os engenheiros entendem outra, o que é um problema por se só mas também uma causa para o problema anterior. O fato de podermos resolver o problema da falta de entendimento sobre o que de fato se pede e o que se entende que foi pedido não elimina o problema da não correspondência entre as características do software entregue e as necessidades atuais do requerente. Pois as divergências entre tais características entregues e estas necessidades atuais do cliente podem ter outra ou outras causas, como , por exemplo, mudanças de tais necessidades dos clientes.

O “como” os engenheiros desenvolvem a solução e a implementam não é o problema principal mas sim uma consequência do possível mal entendimento dos desenvolvedores sobre o que foi pedido, ou da falta de tecnologia , ou falta de processos adequados etc. O que não atende às necessidades é o “como” o software se comporta. Porém a forma como o software é desenvolvido reflete na qualidade dos software. Ramires afirma:

“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).” (Ramires , 2004)

Mas o QFD relaciona o “que se pede” com o “como implementar”. Não vemos , à primeira vista, como isso pode ajudar no problema na falta de entendimento. No entanto, olhando com mais atenção, verifica-se uma relação entre tais aspectos. A relação que se pode fazer é comparativa. Pode - se comparar o conjunto de características que o software teria, se determinado conjunto de ações fosse tomado, com o conjunto de características que o clientes espera que o software tenha. Mas ,

mesmo assim, o conjunto de características que o cliente quer (suas necessidades) é mutável, ao final do projeto provavelmente será diferente das do início do projeto. Assim, pensando dessa forma, pode-se chegar à conclusão de que mesmo que, por meio de validação de requisitos, se chegue a um consenso de que as visões são iguais, o cliente poderá, ao final do projeto não ficar satisfeito com as características do software. Pois as suas necessidades atuais já seriam outras. Ou seja, o conjunto de características do software estariam em desacordo com necessidades do cliente, agora não por causa de falta de entendimento entre o que se pede e o que se entende que foi pedido, mas talvez pela mudança no contexto no qual a organização cliente esta inserida. É bem verdade que tal mudança nas necessidades dariam-se somente se o contexto no qual a organização requerente esta inserida fizesse imposições. Porém não se pode ter certeza de quando o contexto vai impor tais mudanças, se isso vai demorar, se vai ser logo ou se está em fase de transição.

A partir de essa reflexão podemos levantar a seguinte questão: dentre os fracassos nos projetos de software cuja a causa de tal fracasso é atribuída à insatisfação do cliente com as características do software, tal insatisfação tem origem mais na falta de entendimento entre o que se pede e o que se entende que é pedido ou decorre mais de mudanças no contexto (social, econômico, histórico) no qual a organização requerente do software esta inserida?

Também pode ocorrer a seguinte hipótese: o software talvez não correspondesse às necessidades do cliente devido aos conflitos entre requisitos ou restrições que fossem detectados a cada análise feita no conjunto de requisitos durante o processo de desenvolvimento, tais problemas limitariam a forma como o software seria produzido fazendo com que o conjunto de características que o software tivesse não fosse de acordo com as necessidades do cliente.

De qualquer forma, independentemente de quais sejam as causas da insatisfação do clientes, devemos propor abordagens que considerem tais causas.

Sugerimos que haja uma diferenciação entre o significado de necessidades e o significado de expectativas dos clientes. Tal diferenciação que aqui propomos tem o objetivo de facilitar o entendimento do problema da não satisfação do cliente. Então, utilizaríamos o termo necessidades do cliente para fazer referência ao o que de fato o cliente precisa que seja satisfeito, e o termo expectativa referindo-se ao consenso entre solicitante e solicitado sobre o que realmente foi, é e será acordado diante do que se considera viável(lista de requisitos).

Assim poderíamos fazer afirmações tais como “a negociação de requisitos de software é uma atividade tal que tem como fim guiar as tomadas de decisões em situações de conflito visando a satisfação com o maior equilíbrio possível entre as expectativas dos stakeholders e que considera que as necessidades dos clientes mudam a partir de imposições do contexto do qual fazem parte” sem correr o risco de se achar que necessidades e expectativas são as mesmas coisas.

REFERENCIAS

ABRAN, A.; MOORE, J.M., **SWEBOK: Guide to the Software Engineering Body of Knowledge**. IEEE Computer Society, 2004. Disponível em : <<http://www.computer.org/portal/web/swebok/html/ch2#ch2-4.4>>. Acesso em: 20 jul. 2010, 00:30:20.

CAMACHO, Cecília. **Gerenciado Conflitos em Reuniões: Uma estratégia para a Elicitação de Requisitos de Software**. PUC-Rio, 2005. 168p. Dissertação de Mestrado – Departamento de Informática, Potifícia Universidade Católica do Rio de Janeiro.

CARVALHO, A. E. S; TAVAREZ, H.C., Castro, J.B. **Uma estratégia para Implantação de Gerencia de Requisitos Visando Melhoria dos Processos de Softwares**. Centro de Informática – Universidade Federal de Pernambuco, Recife,23p, 2001.

CHICHINELLI, Micheli; CAZARINI, Edson Walmir. **Requisitos não funcionais e sua impotância no processo de desenvolvimento de software**. Encontro Nacional de Engenharia de Produção. Escola de Engenharia de São Carlos – Universidade de São Paulo, São Carlos , 8p, 2001.

Carvalho, E.D., et al., **Negociação e Administração de Conflitos**. Rio de Janeiro : Editora FGV, 2006. 183p.

Couto, Anselmo de Araujo;Martins, Luiz Eduardo Galvão. **Um Processo de Validação de Requisitos Não-Funcionais Baseado no NFR-Framework**.6ª Amostra Acadêmica UNIMEP. Priracicaba. 2008.

Chung L., “Representing and Using Non-Functional Requirements: A Process Oriented Approach” Ph.D. Thesis, Dept. of Comp.. Science. University of Toronto, June 1993. Also tech. Rep. DKBS-TR-91-1.

Chung, L., Nixon, B., Yu, E. and Mylopoulos,J. “Non-Functional Requirements in Software Engineering” Kluwer Academic Publishers 1999.

Cysneiros, Luiz Márcio. **Requisitos Não-Funcionais: Da Elicitação ao Modelo Conceitual**. PUC-Rio , 2001. 224p. Tese de doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Moraes, Renato Oliveira e LAURINDO, F. J. B.. **O método de Monte Carlo com MS Excel**. Bauru: 8 nov. 2004.

Varoto, Ane Cristina. **Visões em arquitetura de Software**. USP, 2002. 108p. Dissertação de Mestrado – Instituto de Matemática e Estatística, Universidade de São Paulo.

Germoglio, Guilherme. **Arquitetura de Software**. 2008. Disponível em <<http://cnx.org/content/m17494/latest/>>. Acesso em: 23 de julho 2009.

Clements, Paul, et al. **Documenting Software Architectures**. 2ed, Addison Wesley, 2002.

GUSMÃO, Critine, et al. **Mapas Conceituais: Utilizando – os na Identificação de riscos de Requisitos**. Engenharia de Software Magazine, Brasil, n 17, ano 2, p. 14-21, 2008.

Leite, J. C. S. P. , **“Engenharia de Requisitos – Notas de Aula”**, 1994.

HAZAN, C., LEITE, J. C. S. P., **“Indicadores para a Gerência de Requisitos”**, Workshop em Engenharia de Requisitos, 2003 WER2003, Piracicaba-SP, Brasil, p. 285-301, 27-28 November.

LEE, W.J., CHA, S.D. & KWON, Y.R. **Integration and Analysis of Use Cases Using Modular Petri Nets in Requirements Engineering**. IEEE Transactions on Software Engineering, vol. 24, no. 12, December, 1998.

NETO, J.S.M. **Integrando Requisitos Não Funcionais ao Modelo de Objetos**. PUC-Rio, 2000. 220p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

PRESSMAN, R. S. **Software Engineering: A Practitioner's Approach**, (Fifth Edition) McGraw-Hill, New York, 2001.

PMBOK, 2004. Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos (Guia PMBOK®) Terceira edição 2004 Project Management Institute, Four Campus Boulevard, Newtown Square, PA 19073-3299 EUA.

RAMIRES, J.J.C.V., **Negociação de Requisitos no Processo de Desenvolvimento de Software**, 2004, p. 199, Dissertação de Mestrado – departamento de Informática – Faculdade de Ciências da Universidade de Lisboa.

RAMIRES, J. e ANTUNES, P. (2004) **Negociação De Requisitos No Modelo Sqfd - Software Quality Function Deployment**. 1ª Conferência Nacional em Interação Pessoa-Máquina - INTERACÇÃO 2004, Lisboa.

SAYÃO, M., Leite, J.C.S.P., **Rastreabilidade de Requisitos**, vol.12,n1,p.30,2005.

SOMMERVILLE, Ian; SAWYER, Peter Wiley. **Requirements Engineering - A good practice Guide** , JONH WILEY & SONS, 1997.

SPÍNOLA, Rodrigo Oliveira, ÁVILA, Ana Luiza. **Introdução à Engenharia de Requisitos**. Engenharia de Software Magazine, Brasil, n 1, ano 1, p. 46-52, 2007.

VASQUEZ, C.E., **Estimativas de Software: fundamentos, técnicas e modelos**. Engenharia de Software Magazine, Rio de Janeiro, n11,p.40-48,2009.

VAROTO, Ane Cristina. **Visões em arquitetura de Software**. USP 2002.108p. Dissertação de Mestrado – Instituto de Matemática e Estatística, Universidade de São Paulo.

Wikipédia, a enciclopédia livre. **Engenharia de Requisitos**. 2010. Disponível em <http://pt.wikipedia.org/wiki/Engenharia_de_requisitos>. Acesso em: 12 de novembro 2010.