

SEBASTIÃO VINÍCIUS ARAÚJO DE MOURA

**NEGOCIAÇÃO DE REQUISITOS DE SOFTWARE : UMA ANALISE DA
VIABILIDADE DE INTEGRAÇÃO ENTRE TÉCNICAS QUE
ESTIMULAM A PARTICIPAÇÃO DO CLIENTE EM NEGOCIAÇÕES
DE REQUISITOS**

Trabalho de conclusão de curso- Monografia apresentada à Coordenação do curso de pós-graduação em engenharia de software do Centro de Ensino Unificado - CEUT, como parte dos requisitos para obtenção do título de especialista em engenharia de software.

ORIENTADOR: Me. FRANCIERIC ALVES DE ARAÚJO

**TERESINA
2011**

SEBASTIÃO VINÍCIUS ARAÚJO DE MOURA

**NEGOCIAÇÃO DE REQUISITOS DE SOFTWARE: UMA ANÁLISE DA
VIABILIDADE DE INTEGRAÇÃO ENTRE TÉCNICAS QUE ESTIMULAM A
PARTICIPAÇÃO DO CLIENTE EM NEGOCIAÇÕES DE REQUISITOS**

Trabalho de conclusão de curso- Monografia apresentada à Coordenação do curso de pós-graduação em engenharia de software do Centro de Ensino Unificado - CEUT, como parte dos requisitos para obtenção do título de especialista em engenharia de software.

Aprovada em: __/__/__

BANCA EXAMINADORA:

Prof.º: Francieric Alves de Araújo

Prof. º: Francisca Neuza de Almeida Farias

1 INTRODUÇÃO

Este trabalho 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. O nome do projeto sofreu uma pequena alteração para que pudesse refletir o seu objetivo. O projeto passou a ter o título de **Negociação de requisitos de software: uma análise da viabilidade de integração entre técnicas que estimulam a participação do cliente em negociações de requisitos de software**. Procuramos demonstrar a partir de argumentos, práticas 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 e incompatibilidade 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.

Estudou-se trabalhos de autores como Ramires (2004), Camacho (2005), Cysneiros (2001), Couto e Martins (2008), Xavier (2009), 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 Kroth (2004), 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 as argumentações e alinhar as idéias específicas com idéias gerais, amplamente aceitas e já legitimadas no meio da engenharia de software. Outras obras além das acima citadas (elas representam a base central deste trabalho) também foram consultadas e encontram-se nas referências e no

decorrer do texto.

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 para que se possa realizar associações às necessidades do processo de desenvolvimento de software.

O capítulo 4, chamado de Verificação de Viabilidade apresenta o resultado da verificação das análises de compatibilidades entre as propostas estudadas; os pontos sugeridos para adaptações; as facilidades e possíveis dificuldades para realizar as adaptações.

O capítulo 5 apresenta as conclusões tiradas durante a pesquisa e também as idéias de trabalhos futuros que podem ser realizados a partir deste trabalho.

2 FUNDAMENTOS

2.1 Requisitos

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 etc. 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, (ÁVILA; SPÍNOLA, 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 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 Ávila e Spínola, 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):

- Identificação.
- Análise e negociação.
- Especificação e documentação.
- Validação.

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

Segundo Ávila e Spínola (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 2.1 representa essa definição.

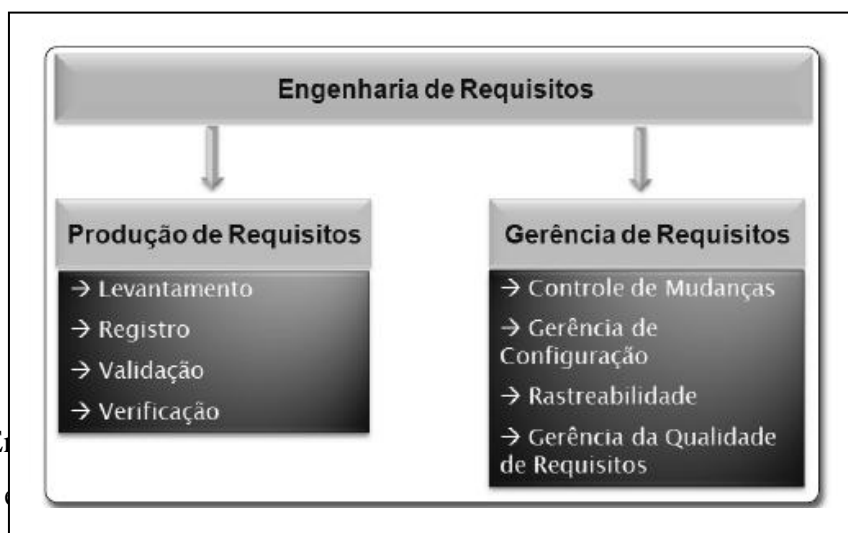


Figura 2.1: Engenharia de Requisitos

Fonte: Ávila e Spínola (2007)

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” (ÁVILA; SPÍNOLA, 2007, p. 49).

É 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 Ávila e Spínola, 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; ÁVILA, 2007, p. 57) é denominada gerência de requisitos.

Segundo Sayão e Leite (2005) existe uma confusão quando se trata de conceitos de Gerência “dos” requisitos e Gerência “por” requisitos. Na visão de Sayão e Leite, Gerência dos Requisitos está associada ao acompanhamento da evolução dos requisitos ao longo do processo de desenvolvimento (manter registro do status do requisito em relação ao desenvolvimento, modificações aceitas e *rationale* associado). E a Gerência por requisitos (geralmente chamada de Gerência de Requisitos) está associado ao controle de todo o processo de desenvolvimento tendo como referência a *baseline* de requisitos.

Ávila e Spínola (2007) citam atividade de gerência de requisitos que devem ser levadas em consideração: *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 que 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 prática 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; 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”.

Esta atividade está muito ligada à próxima atividade, gerência de configuração e se não funcionarem corretamente tanto inviabilizarão as negociações como poderão fazer com que o projeto fracasse.

Gerência de Configuração

A atividade de Gerência de Configuração (GC), segundo Presman:

is a set of activities designed to control change by identifying the work products that are likely to change, establishing relationships among them, defining mechanisms for managing different versions of these work products, controlling the changes imposed, and auditing and reporting on the changes made. (PRESMAN, 2001, p. 225).

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.

Percebe-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 ao mesmo tempo em que trás formalidade. E isto fortalece a comunicação e, conseqüentemente, o relacionamento entre os stakeholders do projeto.

Rastreabilidade

A rastreabilidade, tida como centro das atividades de gerenciamento de requisitos, pode ser entendida como “habilidade de se acompanhar a vida de um requisito em ambas as direções (por exemplo: partindo de requisitos e chegando ao projeto ou, partindo do projeto e chegando aos requisitos) do processo de software durante todo o seu ciclo de vida”(ÁVILA ; SPÍNOLA, 2007, p. 52).

Dependendo do sentido das ligações ou “rastros”, pode-se fazer uso dos termos “pré-rastreabilidade” ou “pós-rastreabilidade”. A pré-rastreabilidade associa os requisitos ao contexto a partir do qual eles surgiram. A pós-rastreabilidade vincula os requisitos ao desenho do sistema e sua implementação (SAYÃO; LEITE, 2007) (DAVIS, 1993).

A figura 2.2 abaixo, extraída de Sayão e Leite (2007), demonstra como as ligações possibilitam acompanhar a vida de um requisito:

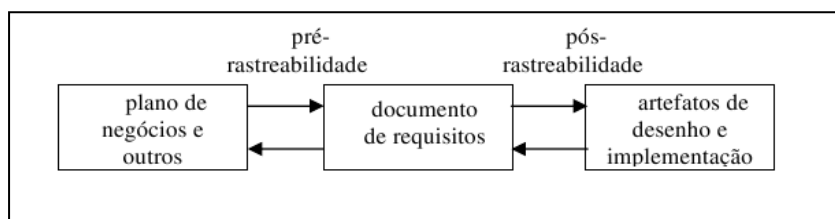


Figura 2.2: Rastreabilidade de Requisitos

Fonte: Sayão e Leite (2005, p.6).

Sayão e Leite comentam a figura:

No primeiro tipo temos a rastreabilidade forward-to (para frente), que liga documentos obtidos no processo de elicitação a requisitos relevantes, e a rastreabilidade backward-from (para trás), que liga requisitos às suas fontes. No segundo tipo temos rastreabilidade forward-from, que liga requisitos a artefatos de desenho e implementação e rastreabilidade backward-to, que liga artefatos de desenho e implementação de volta a requisitos (SAYÃO; LEITE, 2007, p. 7).

Vista como um ponto crítico e difícil de ser realizado, a rastreabilidade geralmente é

feita com a ajuda de ferramentas adequadas para que se torne viável. Algumas ferramentas são citadas em Sayão e Leite como:

- C&L;
- RequisitPro® e;
- DOORS;

Sayão e Leite também mencionam algumas técnicas e classificações.

Uma possível classificação para as técnicas de rastreabilidade mais comuns está relacionada ao foco principal: referências cruzadas ou documentos. Técnicas centradas em referências cruzadas são simples de entender e utilizar e podem ser suportadas pelo uso de hipertexto, esquemas de numeração, indexação ou uso de tags ou ainda por matrizes de rastreabilidade. Técnicas centradas em documentos utilizam modelos (templates) para documentos e artefatos utilizados no processo de desenvolvimento e incorporam mecanismos para transformação e integração de documentos (SAYÃO; LEITE, 2007, p. 17).

Não iremos abordar mais detalhes sobre os tipos de técnicas e classificações que podem ser visto em (SAYÃO; LEITE, 2007).

Gerência de Qualidade dos Requisitos

A gerência de requisitos provê uma infra-estrutura necessária às atividades de verificação, o que possibilita a investigação da qualidade dos requisitos.

O padrão IEEE 830 citam alguns critérios que devemos considerar como: correção, não-ambiguidade, completude, consistência, verificabilidade, modificabilidade (SPÍNOLA; ÁVILA, 2007).

(COUTO; MARTINS, 2008) apresentam um processo para a validação de requisitos não funcionais baseados no framework RNF-framework, que será comentado mais adiante. Este processo pode ser visto como uma forma de gerência de qualidade para requisitos não funcionais, pois visa atender padrões de qualidade baseados, evitando problemas como incompletude, conflitos e ambigüidades, entre outros problemas.

Falhas nestas atividades podem prejudicar bastante as negociações. Pois negociadores estariam utilizando requisitos problemáticos (incorretos, ou desatualizados, ou ambíguos, ou incompletos, ou inconsistentes ou inviáveis – em termos de custo e tempo) para manter suas posições, tomar suas decisões e embasar seus argumentos.

Como os negociadores precisam tratar de assuntos substantivos, assuntos relacionados aos requisitos, se houver algum problema no tocante aos critérios acima citados, os negociadores perderiam tempo em negociações improdutivas e baseadas em dados sem qualidade.

2.3 O léxico ampliado da linguagem

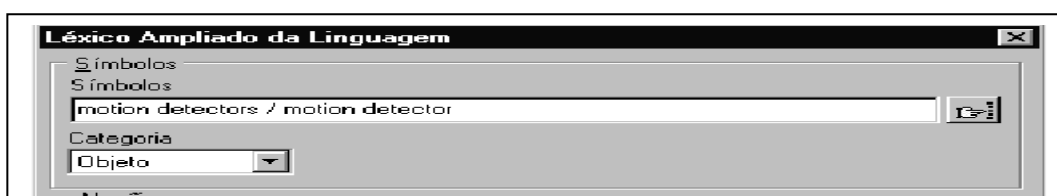
O conceito de *léxico*, de uma forma geral, é o acervo de palavras de um determinado idioma. No contexto do ciclo de desenvolvimento do software e organização para a qual o software será ou esta sendo desenvolvido, o “léxico” pode ser entendido como sendo o acervo de palavras do “idioma de uma organização”. 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 2.3 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 2.3: Exemplo de Entrada do LAL de um sistema de controle de luzes utilizando a ferramenta OORNF.

Fonte: Cysneiros (2001)

Abaixo temos a figura 2.4 mostrando o modelo SADT do processo de construção do LAL. O processo é formado por três atividades : identificar símbolos, classificar símbolos, descrever símbolos.

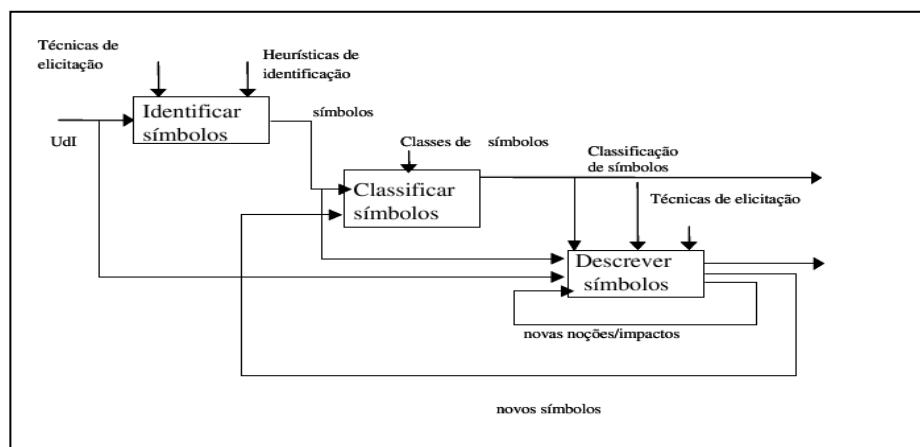


Figura 2.4: Modelo SADT do processo de construção do LAL do UDI.

Fonte: Cysneiros (2001)

2.4 O RNF-Framework

Nesta seção será apresentado o RNF-Framework, feito um esclarecimento sobre conceitos do RNF-Framework e como é feita a geração de grafos em Cysneiros (2001).

2.4.1 O RNF-Framework

No RNF-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, organiza-se metas formando uma estrutura de grafos, no espírito de árvores E/OU que são usados na resolução de problemas (CHICHINELLI; CAZARINI, 2001). No entanto, essas metas raramente podem ser satisfeitas num sentido bem definido. Decisões de projetos podem influenciar de maneira positiva ou negativa em um determinada meta. A avaliação dessas metas determina o grau que um conjunto de requisitos não funcionais esta sendo apoiado por um projeto particular.

O framework consiste de cinco componentes principais, os quais não estão presentes no SQFD (que será abordado mais a frente) (CHICHINELLI; CAZARINI, 2001, p. 4):

- 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 Cysneiros (2004, p:

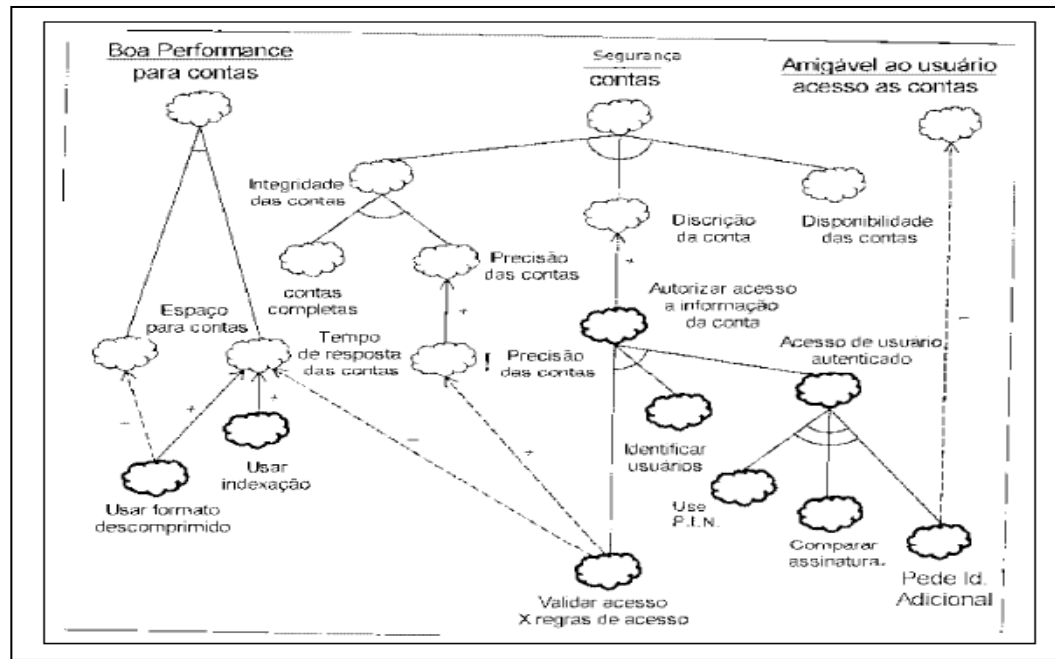


Figura 2.5: Grafo de RNFs decomposto até as Operacionalizações.

Fonte: Chung (2000)

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; 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:

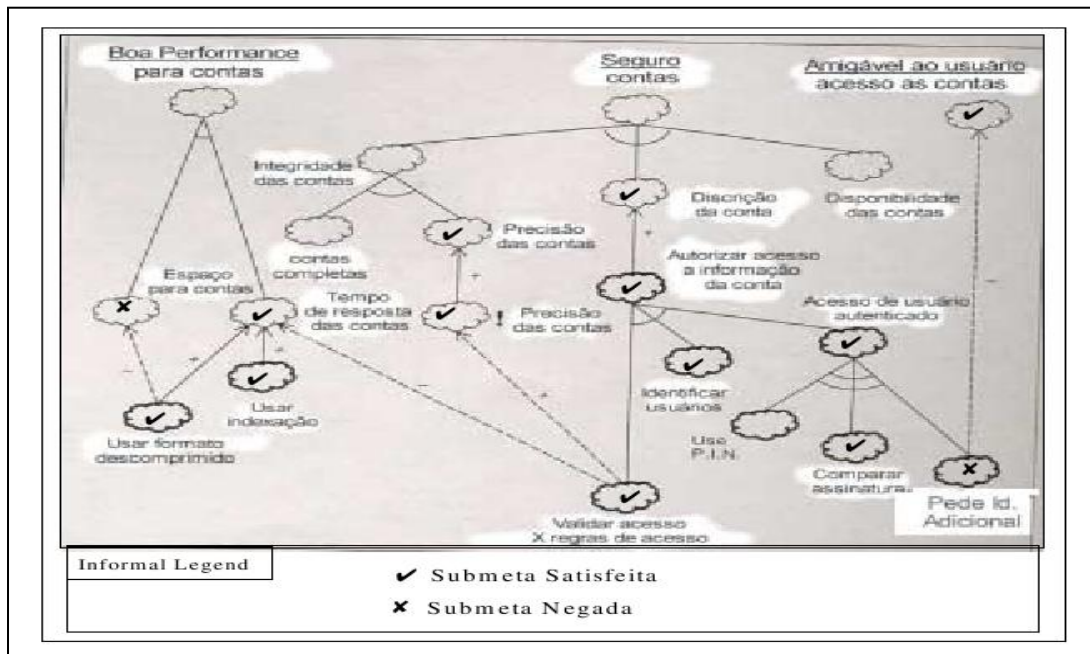


Figura 2.6: Grafo de RNFs com interdependências analisadas.

Fonte: Cysneiros (2001)

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

2.4.2 Esclarecendo Diferenças entre, RNF, meta, Operacionalização, tópico e tipo

Julga-se necessário fazer uma diferenciação entre os conceitos de RNF, meta, operacionalização, tópico e tipo no contexto de grafos de RNF para que se consiga mais fluência para prosseguir com o assunto.

O primeiro ponto a ser esclarecido é o conceito de Requisito Não funcional. Os requisitos não funcionais representam as restrições ou atributos de qualidade para um software ou para o processo de desenvolvimento do software. Podemos citar segurança, precisão, usabilidade, performance, portabilidade, manutenibilidade como exemplos de classes ou sub-classes (tipo ou tópico, de acordo com o framework RNF) dentro de uma

taxonomia nas quais um determinado requisito não funcional poderia ser enquadrado. Dois exemplos de RNFs que se enquadram nas classificações portabilidade e performance, respectivamente, são citados a abaixo:

“O software deve ser compatível com os browsers IE (versão 5.0 ou superior) e Firefox (1.0 ou superior)”;

- a) “o software deve garantir que o tempo de retorno das consultas não seja maior que 5 segundos”;

Alguns autores defendem que os requisitos devem ser apresentados seguindo uma qualificação ou quantificação (GILB apud RAMIRES, 2004). Pode ser observado que isso se verifica nos exemplos supracitados. O primeiro e o segundo exemplos (itens a e b) expressam, de forma não vaga ou não ambígua, qualidades que se deseja que o software tenha.

Um outro ponto, é que na representação de Chung os requisitos não funcionais são encapsulados na forma de **metas** a serem satisfeitas, o que dá um aspecto de ambiguidade. Isto contrasta com idéias de alguns autores como Gilb e Ramires que defendem a idéia que falsos requisitos devem ser evitados. Falsos requisitos, segundo (RAMIRES, 2004, p. 26) “ocorrem quando uma especificação contém elementos desnecessários”. Ramires faz referência a Gilb que explana situações onde ocorrem os falsos requisitos:

é comum encontrar requisitos como custos reduzidos, promover a gestão de controle, elevada usabilidade e aumentar a satisfação do cliente, onde: não existe uma clara definição dos objectivos que se pretendem; não existe recurso a um conjunto de termos de base que reduzam ambiguidades; não existe quantificação do nível de qualidade corrente nem do nível exigido de futuro (RAMIRES, 2004, p. 26).

Na proposta de grafos de RNFs (CHUNG, 2000)(CYSNEIROS, 2001), *custos reduzidos*, *promover a gestão de controle*, *elevada usabilidade*, e *aumentar a satisfação do cliente* já seriam considerados necessários. Seriam classificados como metas a serem atingidas e seriam decompostas até se chegar às operacioanalizações. “Uma operacionalização corresponde a ações ou atributos que claramente identifiquem o que é necessário para satisfazer a meta principal (CYSNEIROS, 2001, p. 37).

Outro conceito a esclarecer é sobre as formas ou critérios de decomposições para os RNFs no Framework RNF. São elas: a por tipos e a por tópicos. Cysneiros discorre sobre as formas de decomposição de RNFs, conceitua a decomposição por tipos e por tópicos e dá exemplos dessas decomposições. Na decomposição por tipo os métodos de decomposição existentes no framework são usados como fundamentos e guiam os engenheiros para as decomposições possíveis de uma determinada meta. “No caso do RNF Segurança, por exemplo, integridade, confidencialidade e disponibilidade podem ser possíveis decomposições desta meta” (CYSNEIROS, 2001, p.).

Percebe-se que Segurança expressa acima funciona como uma classe dentro de uma taxonomia, tal como foi comentado anteriormente no início dessa seção, e que integridade, confidencialidade e disponibilidade podem ser percebidas como sub-classes (ou tipos, de acordo com a terminologia utilizada por Chung (2000), Cysneiros (2001)) de Segurança.

Outra forma de decomposição é a por tópicos: “A decomposição por tópicos trata da decomposição estrutural do problema. Por exemplo, no exemplo acima, o RNF Seguro aplicado a contas poderia ter sido decomposto em contas pessoa física e jurídica” (CYSNEIROS, 2001, p.38).

Assim, resumindo, pode-se ver que tópico (assim como tipo) é uma forma ou critério de decomposição de RNF; meta é a forma de concepção para RNF que deve ser vista como um objetivo a ser alcançado; operacionalização é uma ação ou atributo, indica o que é necessário para atingir a meta principal ou parte dela (ou seja, é a sub-meta imediatamente superior à operacionalização em questão) e; RNF é uma restrição ou atributo de qualidade.

Analisando as relações entre os conceitos verifica-se que quando atributos de qualidade são decompostos, seguindo o critério de tópicos, chegamos a alternativas de solução (operacionalizações) que podem ou não impor uma restrição à satisfação de outras metas. Pode-se, então, identificar interdependências entre os RNFs. Estas interdependências são identificadas com maior facilidade quando se chega às operacionalizações. E podem ser do tipo positivas ou negativas. As positivas ajudam na satisfação das outras metas. As negativas impõem restrições (uma restrição pode ser total ou parcial) na satisfação de outras metas. Tais metas têm origem nos seus respectivos atores do UDI. Portanto quando uma operacionalização tem interdependência negativa com metas cuja a origem é outro ator do UDI temos os conflitos de interesses, ocasionando, assim, uma necessidade de negociação

para a priorização das metas.

2.4.3 Geração de Grafos de RNFs

Cysneiros (2001) propôs a utilização do grafo de RNFs apresentado em 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 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 uma figura extraída de (CYSNEIROS, 2001) demonstrando as origens Administrador do Prédio e CIPA para os RNFs Custos Operacionais e Seguro, respectivamente.

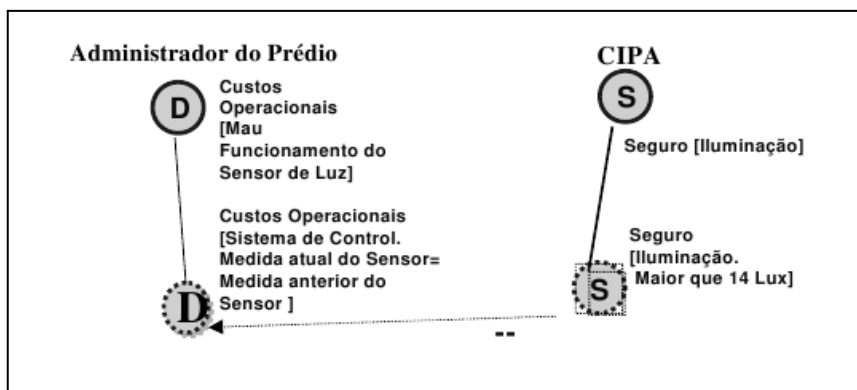


Figura 2.7: Exemplo de Grafo de RNFs com Identificador de Origem do RNF

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:

A Figura 2.8 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 (CYSNEIROS, 2001).

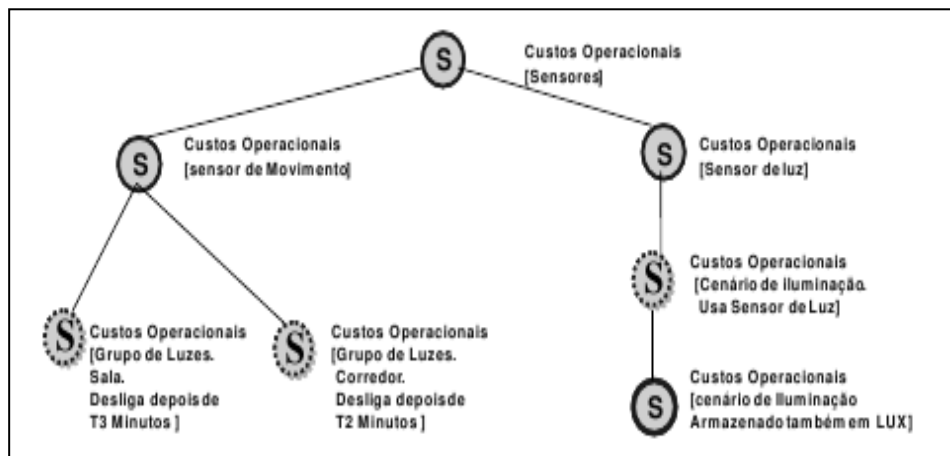


Figura 2.8: Exemplo de Operacionalizações Estáticas e Dinâmicas

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) (CYSNEIROS, 2001).

A seção 2.7 da tese de Cysneiros trata sobre conceitos básicos relativos aos grafos de RNF os quais já foram introduzidos anteriormente neste texto.

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
estática	nem dados / nem ação	n/a	num sistema de informações laboratorial, poderemos encontrar um RNF Usabilidade aplicado a Listas cuja operacionalização estabelece a		

Figura 2.9: Classificação das metas.

Quanto a *representação escrita para as metas*, a Figura 2.10 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 decomposição de metas e as opções para a representação das metas.

META	TÓPICO	REPRESENTAÇÃO 1 (COMPLETA)	REPRESENTAÇÃO 2 (SIMPLIFICADA)
<i>Custos Operacionais</i>	<i>Sensores</i>	<i>Custos Operacionais [sensores]</i>	<i>igual a primeira</i>
	<i>Sensor de Movimento</i>	<i>Custos Operacionais [sensores. Sensor de Movimento]</i>	<i>Custo Operacional [Sensor de Movimento]</i>
	<i>Sensor de luz</i>	<i>Custos Operacionais [sensores. Sensor de luz]</i>	<i>Custo Operacionais [Sensor de luz]</i>

Figura 2.10: Quadro de representação escritas das metas.

Fonte: do Autor

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 (adaptado):

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 7(CYSNEIROS, 2001).

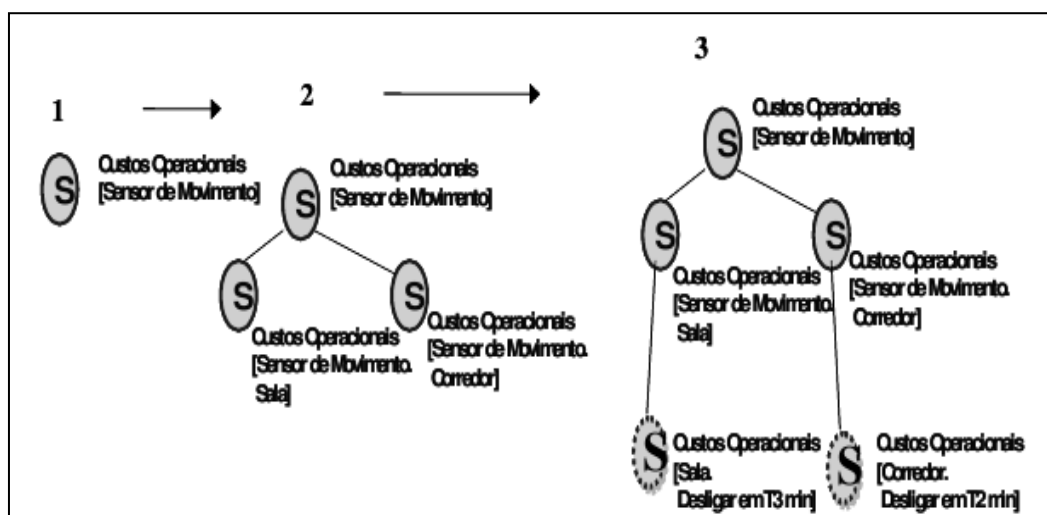


Figura 2.11 – Exemplo da Construção do Grafo de RNFs

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.

2.5 Análise 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 (apud Xavier (2009)), 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 os 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 referência a Bittencourt ao explicar que as informações utilizadas para especificação de um sistema de informação vêm dos elementos de negócio:

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, p.9).

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 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 BPMNRF

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

O SQFD é uma metodologia de desenvolvimento de software que fornece métodos formais para transformar os requisitos de qualidade em especificações de desenho. O SQFD (Software Quality Function Deployment – SQFD) é uma extensão do Desdobramento da Função Qualidade (Quality Function Deployment – QFD) aplicada ao processo de desenvolvimento de software.

O QFD tem como objetivo a identificação dos stakeholders que pretendem o sistema, a identificação do que desejam e a identificação de como satisfazer esses desejos.

A figura a baixo exhibe uma matriz simplificada do QFD:

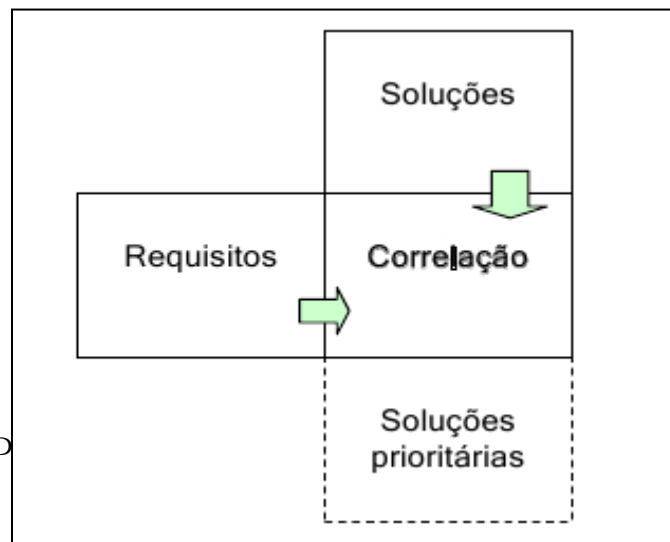


Figura 2.12: Matriz QFD
Fonte: Ramires (2004).

No Capítulo 3 será comentada mais sobre o QFD e SQFD.

2.7 Integração visão funcional com visão não-funcional

Luiz Márcio Cysneiros (Cysneiros, 2001), em sua tese intitulada de “Requisitos Não-Funcionais: Da elicitação ao modelo Conceitual”, apresenta alguns objetivos a serem atingidos em seu trabalho:

- Desenvolver processo para se lidar com requisitos não-funcionais desde as etapas iniciais do processo de desenvolvimento de software.
- Obter uma melhoria da qualidade final do produto sob a ótica, tanto do cliente quanto do desenvolvedor, bem como em um menor tempo de entrega de produtos e menores custos de manutenção. Para isso buscou atingir um outro objetivo que é:
- Obter modelos conceituais mais completos e com os impactos da satisfação dos requisitos não funcionais já analisados. Para este ser atendido, por sua vez, verificou que era necessário atingir outro objetivo:
 - Obter um modelo conceitual que enfoque os requisitos funcionais e não funcionais ao mesmo tempo.

Para que esses objetivos fossem alcançados Cysneiros propõem uma estratégia que se baseia no uso Léxico Ampliado da Linguagem (LAL) para que sirva como âncora da construção dos modelos funcionais e não funcionais.

Elaborou-se uma extensão ao LAL para que ele pudesse sustentar a natureza evolutiva e as negociações da elicitação dos RNFs. A ferramenta OORNF (Neto 00) foi customizada para suportar as novas características dadas ao LAL. Assim o engenheiro de software poderia administrar melhor tal aspecto evolutivo dos RNFs.

Outro aspecto dos RNFs, o do tratamento de interdependências entre RNFs, teve bastante atenção em seu trabalho. Estes por terem a possibilidade de originar importantes decisões de desenho e que “implicam em grande montante de retrabalho se não forem prévia e convenientemente identificadas” (Cysneiros, 2001, p.13.).

Mais uma contribuição proposta pelo autor foi uma adaptação do grafo de RNFs proposto por Chung (Chung 2000). O objetivo primário dessa adaptação era o de obter um certo nível de rastreabilidade dos RNFs para auxiliar em seu aspecto evolutivo. Outro objetivo é prover facilidade de integração entre as visões funcional e não-funcional.

Cysneiros (2001) mostra ainda como organizar os grafos de forma que possibilite que a realização da análise sistemática das interdependências intra-grafos seja facilitada. Uma proposta para a integração das visões funcional e não funcional com o fim de obter um modelo conceitual consolidado refletindo ambas as visões também foi apresentada.

Por fim foi proposto uma extensão a UML nos diagramas de classe, sequencia e colaboração. O objetivo é representar as consequências das operacionalizações dos RNFs e

estabelecer uma ligação dos modelos da visão funcional com os RNFs que possibilite a rastreabilidade desses RNFs. Essa rastreabilidade auxilia bastante o tratamento do aspecto evolutivo do software, pois podemos nos reportar às fontes que dão origem a inclusão nos modelos conceituais de uma determinada classe, operação, atributo ou mesmo mensagem vinda da operacionalização de um RNF (CYSNEIROS, 2001).

Seu trabalho se enquadra sob a ótica de processo de desenvolvimento que propõem formas de justificar decisões sobre inclusão ou exclusões de requisitos que refletirão no desenho do software e que também guia, racionalizando em termos de RNFs, o processo de desenvolvimento de software.

Cysneiros ressalta um aspecto dessa ótica que é o fato de tomadas de decisões serem forçadas quando ocorre adição de RNFs a uma especificação de requisitos. Isso poderá provocar um efeito positivo ou negativo em outros RNFs e estes efeitos podem ser visualizados através de *interdependências entre RNFs*. Um RNF pode afetar um outro RNF de forma positiva ou negativa. A positiva ajuda na satisfação do requisito. E a negativa força uma tomada de decisão.

A estratégia formulada por Cysneiros (2001) pode ser visualizada, visão geral, na figura abaixo:

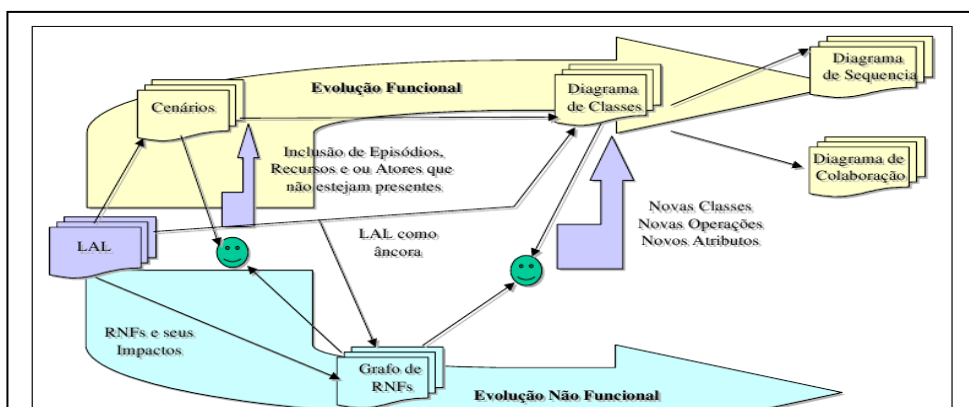


Figura 2.13: Detalhamento da estratégia proposta por Cysneiros.

Fonte: Cysneiros (2001).

A estratégia geral tem como objetivo: demonstrar como identificar e representar RNF de maneira geral e integrar as visões funcional e não funcional. Também propõem o processo de elicitación que busca a sistematização para obtenção, representação e análise de RNF, possibilitando a análise sob duas óticas: interdependências um RNF e outro(s) RNF(s) e interdependências entre RNF(s) e Visão Funcional.

A estratégia impõe algumas restrições como a construção de diagramas de classe e grafos que devem ser realizadas a partir de símbolos de LAL (os nomes dos grafos e diagramas de classes tem que ser um símbolo do LAL) e o LAL será fonte de definição de **cenários**¹. O motivo das restrições esta na facilidade de análise de impacto do RNF no modelo conceitual.

Na proposta a evolução dos requisitos é vista sobre dois ciclos distintos que é o funcional e não-funcional. E o LAL não esta classificado em nenhum deles, é a ancora ou base do dois.

O tratamento do aspecto evolutivo dos RNF foi favorecido pela extensão realizada no LAL que foi a adição, no LAL do UDI, de necessidade de RNFs e dos seus impactos sobre os símbolos. A integração das visões foi favorecida pela inclusão de RNF no modelo funcional. A integração é feita ao nível de cenários ou de diagramas de classe, sequência e colaboração. Ao nível de cenários deve-se avaliar o título do cenário e procurar relação com o grafo. Ao nível de diagrama de classe deve-se, para cada classe, procurar a ocorrência do símbolo que nomeia a classe no conjunto de grafos. Os dois ciclos evoluem independentemente. Sempre que ocorrer uma alteração em qualquer das visões, o engenheiro deve atualizar a outra visão e fazer a integração da alteração na outra visão.

Quando for detectado um novo RNF ou condição de satisfação, deve-se analisar os impactos na visão não-funcional e integrá-lo à visão funcional. Deve-se analisar os cenários que se relacionam com os RNF alterados e o mesmo deve ser feito com os

¹ O termo Cenário pode ser entendido como descrição de situações que ocorrem no macro-sistema e suas relações com o sistema.

diagramas de classes.

Quando a mudança ocorre na visão funcional ocorre o processo inverso: faz-se a atualização do LAL, dos cenários, diagramas de classes, de sequência e de colaboração. No caso de uma nova classe for incluída, inicia-se o processo de integração para essa nova classe e verifica-se se as novas funções não demandam novos RNFs (se isso ocorrer realiza-se novamente a integração do RNF na visão funcional).

Um processo de construção para a visão Não-Funcional pode ser visualizado na figura abaixo:

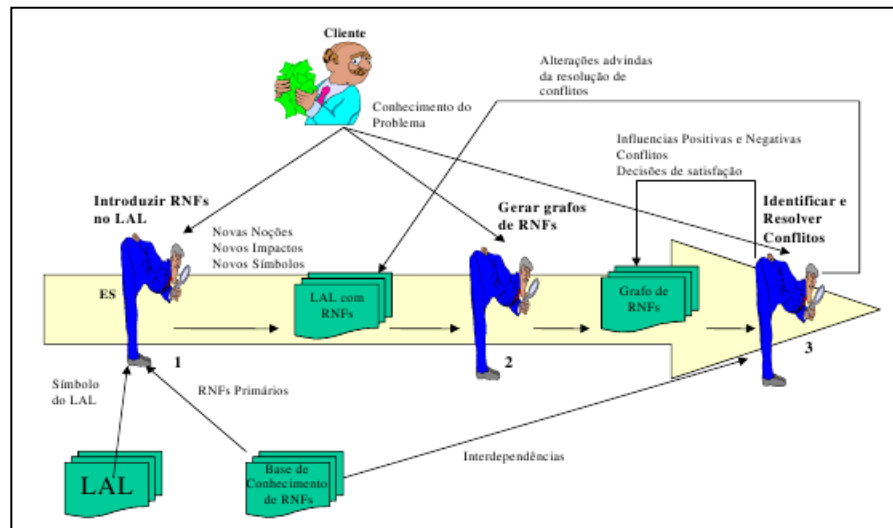


Figura 2.14: Processo de elicitação de Requisitos Não-Funcionais

Fonte: Cysneiros (2001)

2.8 Considerações finais

Neste capítulo foram apresentados conceitos de engenharia de requisitos considerados básicos para o entendimento da análise a ser apresentada no capítulo 4. Mais conceitos básicos serão apresentados no capítulo 3 se referindo especificamente à negociação de forma geral e negociação no processo de desenvolvimento de software.

3 NEGOCIAÇÃO DE REQUISITOS DE SOFTWARE

3.1 Aspectos Conceituais

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. Aqui nesta seção (3.1), fala-se 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 este 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 Carvalhal 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 esta 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 idéia, um caminho de ação, um maneira de punir, um critério para dividir lucros etc. O objeto de negociação caracteriza o que esta 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) (Carvalhal 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

Aqui nesta seção falar-se-á brevemente sobre os tipos e causas de conflitos. Para que haja negociação é necessário a ocorrência de conflito. O conflito se torna, de certa forma o ponto central da negociação, ou seja a razão de haver negociação. 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 causam impacto nas 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 tipos interpessoais e intergrupais.

Robbins (apud Camacho, 2005) distingue 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 exercício das atividades 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.

Os conflitos entre requisitos (sejam do tipo funcionais com não funcionais ou até não-funcionais com não-funcionais) geram conflitos de interesses entre os stakeholders da mesma forma que os conflitos entre os requisitos e recursos. O trabalho de Cysneiros (2004) prevê uma estratégia de tratamento dos requisitos. Em seu trabalho o LAL serve como ancora para o desenvolvimento e integração entre as visões funcional e não funcional do software, utilizando, para isso, o, UML, OCL e o RNF- Framework. No trabalho, é possível a

identificação e exposição desses tipos de conflitos.

Conflitos nos requisitos podem causar momentos de tensão na equipe de desenvolvimento e também entre a equipe e os demais interessados no software.

Tais momentos de tensão decorrem da expectativa do envolvimento das pessoas em situações conflituosas (Carvalho, 2007). Como foi visto anteriormente, uso de RNF-Framework é uma maneira bastante eficiente de expor conflitos entre requisitos, o que pode ser usado na exposição e documentação de tais situações e das decisões tomadas, contribuindo para ajudar a diminuir a ansiedade dos participantes em entender os conflitos e visualizar suas relações.

3.3 Abordagens de Negociação

As abordagens de negociação podem ser de 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*).

Este trabalho se apóia em abordagens práticas (Ramires, 2004). O processo proposto por Ramires se enquadra entre as abordagens práticas e é um misto entre as abordagens prática integrativa e abordagem prática distributiva.

3.4 Aspectos Relacionais e Substantivos da Negociação

“Todo negociador tem dois tipos de interesse: na substância e na relação” (Fisher, Ury, Patton apud Carvalho, 2007). É fundamental desenvolver a habilidade de identificar 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 Carvalho 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 deve ser detalhada em estágios e cada estágio ser 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 pós-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 do SQFD (mas o processo a aplicar nas demais áreas é análogo). 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 visa o auxílio à resolução de conflitos de *corr*² entre stakeholders e a obtenção de um valor de consenso. O sistema também propõe um valor baseado em voto maioritário e comportamento distributivo.

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 3.1:

² Valor numérico que representa o interesse do negociador em relação a uma correlação requisito/solução

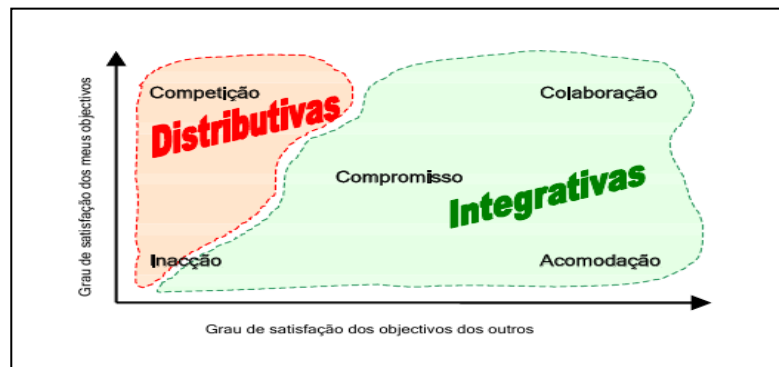


Figura 3.1: Atitudes Integrativas e Distributivas
 Fonte: Ramires (2004).

A partir da análise da figura verifica-se que 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. Percebeu-se neste trabalho que as matrizes do SQFD podem ser substituídas pelo RNF-Framework ou até utilizadas em conjunto.

Máquina 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

$$\text{CÉLULA} = \{ \{E \mid T\}, \{ \text{corr}, \text{ScÉLULA} \} \}$$

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

Se no estado $E(b)$ ⁵ 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 $T(n)$ ⁶. Senão, mantém-se o estado $E(b)$ ⁷.

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 ($\exists NÃO_CONCORDO$), retorna-se ao estado $T(n)$.

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

Por outro lado, se um stakeholder S_i tiver uma atitude INACAODIST ($INACAODIST(S_i)$) 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 INACAODIST 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 INACAODIST ($INACAODIST(S_i)$) 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*

⁵CÉLULA = $\{E(b), \{\{corr(x), S_i\} \mid \{\{corr(x), S_i\}, \{corr(y), S_j\}, x=y\}\}\}$

⁶ $E(b) \rightarrow x \neq y \rightarrow T(n)$

⁷ $E(b) \rightarrow x=y \rightarrow E(b)$

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 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 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 INACÇÃO_{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:

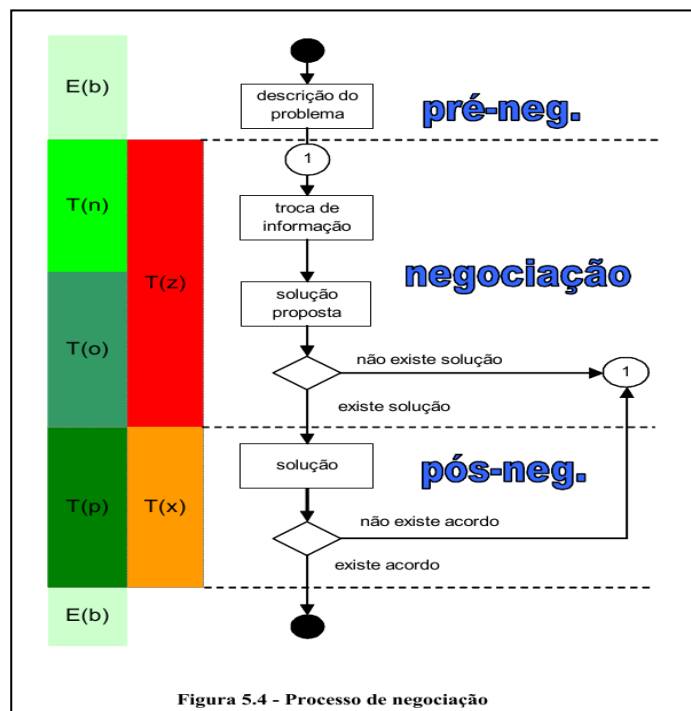


Figura 3.3: Processo de Negociação
Fonte: Ramires (2004)

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.”

3.5 Considerações Finais

A negociação de requisitos deve ser realizada de forma lúcida durante toda a fase de engenharia de requisitos. É essencial para a racionalização do processo de negociação de requisitos que se entenda claramente os conceitos de *Necessidades e interesses*, *Objeto e Escopo*, *Objetivos e Posições*, *Moeda de Troca*, *Argumentos*, *Pontos de recuo* e *MACNA* e que se faça o seu mapeamento para o contexto de um determinado projeto de desenvolvimento de software. Como foi exposto neste capítulo o sucesso nas negociações, ou seja, a boa qualidade das negociações provocarão um efeito positivo na boa participação dos stakeholders e isso revela a importância da formalização das negociações.

A proposta apresentada por Ramires abre uma boa oportunidade para que se consiga aumentar a agilidade e racionalidade das negociações de requisitos em um projeto de desenvolvimento de software. Porém pode ser melhorada fazendo-se com que ela considere os principais conceitos apresentados nesse capítulo e que permita aos negociadores perceberem que estão considerando tais aspectos.

No próximo capítulo será apresentado o capítulo que fala sobre a verificação de viabilidade de integração entre propostas que auxiliam a negociação de requisitos e pesquisa sobre a possibilidade de desenvolvimento de uma metodologia de negociação de requisitos.

4 A ANÁLISE DE VIABILIDADE DE INTEGRAÇÃO

4.1 A estratégia de pesquisa, critérios de triagem bibliográfica e análise de compatibilidades

Para que se pudesse chegar ao conjunto final de propostas a serem estudadas e analisadas, um levantamento bibliográfico inicial foi realizado. Um critério estratégico de levantamento bibliográfico utilizado para esta pesquisa foi o de palavras-chave em torno do assunto principal, como restrições, qualidade, tempo, negociação, e requisitos não-funcionais.

O assunto principal definido foi negociação de requisitos de software. Percebeu-se, durante a execução do projeto de pesquisa que seria inviável identificar e listar todas as compatibilidades existentes entre os trabalhos levantados visto que são vários, como se pode comprovar no projeto, e muito complexos. A metodologia apresentada no projeto não previa um processo de comparação com passos, ferramentas e lista de critérios que ajudassem a atingir tal objetivo dentro do prazo do projeto de pesquisa⁸. Outro problema encontrado durante a execução do projeto foi a volatilidade da bibliografia. A execução do projeto deparou-se com o surgimento natural de novas referências bibliográficas e eliminação de outras à medida que se foi aprofundando no tema. Devido a esses contratempos, optou-se por definir critérios para realização de uma triagem entre esses trabalhos, durante a execução do projeto, para depois realizar uma análise mais apurada do material resultante. Para essa análise foi desenvolvido um método, que é apresentado no apêndice A.

Como as negociações em geral só se iniciam quando algum conflito é percebido procurou-se por formas de exibição de conflitos, o que levou à descoberta do RNF-Framework, e a partir daí passou-se a procurar por suas ocorrências dentro da bibliografia inicialmente levantada. Este passou a ser o primeiro critério utilizado na triagem do conjunto bibliográfico apresentado no projeto. Os demais critérios são: *complementaridade de idéias e complementaridade de tecnologias e seus métodos de uso*. Depois da realização da triagem, aplicando-se esses critérios, chegou-se ao conjunto final com quatro trabalhos a serem analisados.

4.1.1 Considerações a respeito dos critérios de triagem

⁸ O projeto de pesquisa sofreu alguns refinamentos em seu texto e título, porém as premissas e objetivos gerais foram mantidos do início até o fim do projeto. Ele é apresentado no Apêndice B deste trabalho já com os ajustes.

Como comentado anteriormente, o primeiro ponto que influenciou a triagem foi a forte presença do RNF-framework. Dentre os quatro trabalhos que foram selecionados pela triagem, três trabalhos são baseados no framework. O único trabalho que não fazia referência ao framework é o de título de Negociação de Requisitos no Processo de Desenvolvimento de Software de Ramires (2004). Percebeu-se durante a pesquisa que as matrizes do SQFD utilizadas no trabalho de Ramires poderiam ser substituídas pelos grafos de RNFs ou usadas em conjunto com eles, por isso seu trabalho não foi descartado. Além disso, a sua proposta se tornou o fio condutor desta pesquisa, pois foi a única referência a nível de tese⁹ encontrada durante a fase de coleta bibliográfica.

O segundo critério considerado foi o da complementaridade entre as propostas ou *complementaridade de idéias*. Percebeu-se durante a pesquisa que o processo de negociação de requisitos (RAMIRES, 2004) se dá dentro do contexto de processo de engenharia de requisitos (DIDIER, 2003) e que este ocorre dentro do contexto de processo de desenvolvimento de software (PRESMAN, 2001) que, por sua vez existe para produzir software para auxiliar processo(s) de negócio (ALLWEYER, 2010) (XAVIER, 2009). Desta forma a triagem feita entre os trabalhos da bibliografia levantada foi também influenciada levando-se em consideração essa *complementaridade de propostas* dentro dessa hierarquia de processos. Durante esta pesquisa, antes mesmo da leitura de (XAVIER, 2009) percebeu-se o grau de importância que um bom entendimento do negócio e dos processos de negócios têm para a engenharia de requisitos. Ao mesmo tempo percebeu-se que a engenharia de requisitos feita sem levar em consideração o negócio impacta negativamente na qualidade que o software precisaria ter para satisfazer o negócio. Apesar de que cada proposta se destinava a um fim específico notou-se que há idéias em comum entre as propostas como: a necessidade de que haja consenso entre os participantes a cerca dos requisitos, a premissa de que as decisões de projeto devem ser tomadas antecipadamente à implementação do software, a importância da boa comunicação e participação dos stakeholders no processo, etc.

Também, outro ponto que influenciou a escolha foi a presença de outras tecnologias e seus métodos de uso. Este critério é o mais complexo e trabalhoso pois obriga a observação da forma como as várias tecnologias citadas em cada propostas são usadas para suportar as propostas e como beneficiariam o processo de negociação. Este critério é mais relacionado com a análise de compatibilidades entre as propostas realizada no Apêndice A.

⁹ Esta é uma restrição de qualidade explicitada, na metodologia do projeto de pesquisa, para a execução da fase 1 do projeto.

4.1.2 O resultado da triagem

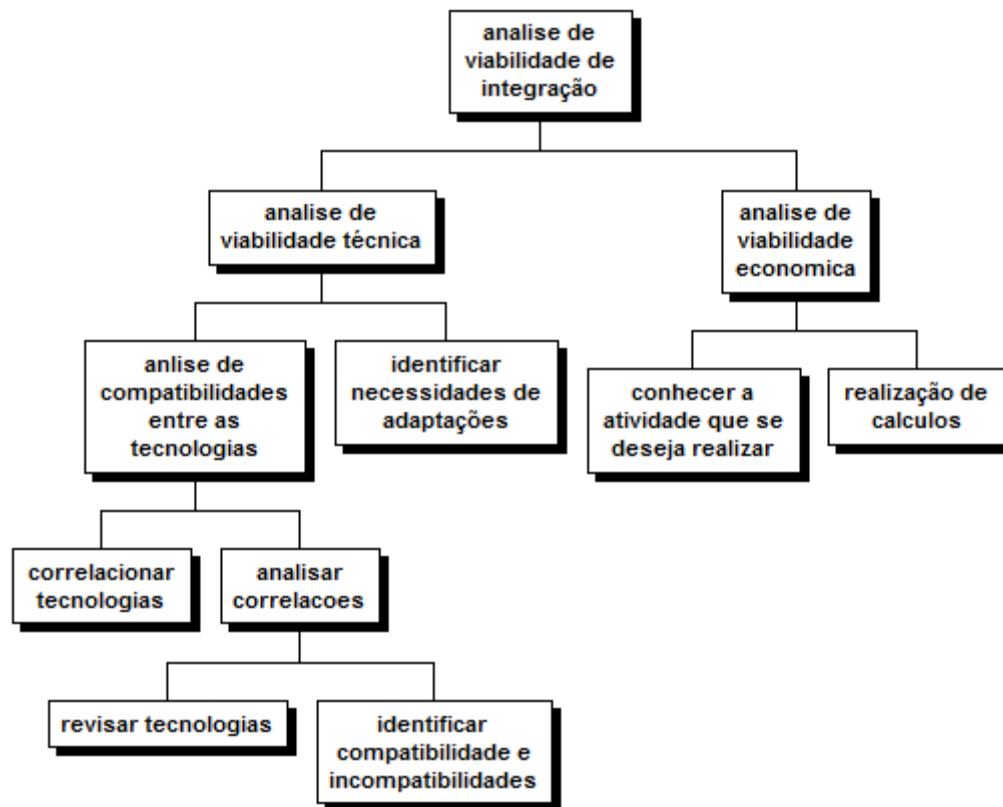
Após realização da triagem, chegou-se á lista de trabalhos abaixo:

- Estratégia de elicitação e integração de RNFs ao modelo conceitual do software de Cysneiros em “Requisitos Não-Funcionais: Da Elicitação ao Modelo Conceitual” (CYSNEIROS, 2001).
- O processo de Engenharia de Requisitos proposto por Didier em “WRE-Process: Um Processo de Engenharia de Requisitos Baseado no RUP” (DIDIER, 2003).
- O processo de negociação de requisitos proposto por Ramires em “Negociação de Requisitos no Processo de Desenvolvimento de Software” (RAMIRES, 2004)
- A proposta e Integração de RNFs com o BPM proposta por Xavier em “Integração de Requisitos Não-Funcionais a Processos de Negócio: Integrando BPMN e RNF” (XAVIER, 2009).

Definido o conjunto final de trabalhos, este foi submetido à aplicação da análise de viabilidade de integração.

4.2 A análise de viabilidade de integração

A análise de integração entre as propostas estudadas compreende duas análises específicas: análise de viabilidade técnica e análise de viabilidade econômica. Uma análise sobre este último critério não foi realizada nesta pesquisa devido à sua especificidade e dependência de estabelecimento de um contexto econômico e temporal dentro de uma organização ou projeto. Abaixo é exibido o gráfico WBS (Work Breakdown Structure) do trabalho que seria necessário pra a realização da análise de viabilidade de integração entre as propostas.



Apenas a análise de viabilidade técnica de integração foi realizada durante a execução desta pesquisa.

4.2.1 A Análise de compatibilidades entre os trabalhos selecionados

Para garantir a qualidade dos resultados, facilidade e agilidade na análise de compatibilidades/incompatibilidades, foi desenvolvido um método.

A seguir são apresentadas convenções sobre o que é considerado compatibilidade e incompatibilidade, bem como a exposição das próprias compatibilidades e incompatibilidades identificadas durante a aplicação do método de análise de compatibilidade do apêndice A.

4.2.1.1 Conceitos de Compatibilidades e incompatibilidades

Primeiramente, considera-se compatibilidade a concordância ou alinhamento de idéias e facilidade de integração entre tecnologias que se apresentam complementares. Se duas propostas fazem referência a uma mesma notação ou notações que contêm características que se somam e são fáceis de integrar formando um conjunto tecnológico mais abrangente e poderoso, por exemplo, essas propostas nesse ponto de vista são compatíveis. Se duas propostas apresentam idéias ou tecnologias diferentes uma das outras e isso provocar conflito ou dificuldades de uso em conjunto das propostas caracteriza-se incompatibilidade. Se duas propostas não citam nenhuma referência a determinada tecnologia ocorre uma lacuna. Nesses dois últimos casos percebe-se que tanto na presença de tecnologias conflitantes quanto na ausência de tecnologias há dificuldades. No caso de presença de conflito depara-se como o retrabalho, ou seja a migração de uma tecnologia para outra com o fim de eliminar a heterogenidade de tecnologias. Isso pode representar em custos maiores que a criação de uma adaptação.

4.2.1.2 Aplicação do método para identificação de compatibilidade e incompatibilidade

Como resultados da aplicação do método foram obtidas os seguintes resultados:

- **Incompatibilidades**

Durante a análise técnica realizada nas propostas, seguindo o critério de compatibilidade/incompatibilidade verificou-se que fazendo algumas adaptações se poderá ter um alto alinhamento entre as 4 propostas analisadas. Foram encontradas mais compatibilidades do que incompatibilidades. Somente quando se tratou de linguagens de programação utilizadas para a implementação das soluções as incompatibilidades, segundo o conceito de incompatibilidade utilizado nesta pesquisa, foram encontrados.

Como casos que chamaram atenção, foram encontrados UML com BPMN e RNF-Framework com SQFD que embora em parte se destinassem ao mesmo fim foram vistas como mais complementares do que concorrentes.

- **Compatibilidades**

O RNF – Framework é compatível com a proposta de Ramires e a metodologia SQFD, podendo-se utilizá-los em conjunto e servir de meio para integrar a proposta de Ramires às propostas de Didier, Cysneiros e Xavier para apoiar as negociações de conflitos visando a lucidez das negociações considerando-se processos de negócios, e processo de engenharia de requisitos (elicitação, análise de requisitos).

O LAL viabiliza a formação de uma ontologia comum entre os stakeholders nas propostas de Xavier, Didier, Cysneiros e Ramires;

A racionalização das negociações na proposta de Ramires que é baseada no sistema IBIS pode ser sustentada e acelerada pelo a adoção do LAL e também pela Técnica de Toranzo.

O LAL serve de ancora para formação das visões funcional e não funcional do software e também pode ser usado como ancora para definir as visões de negócio, permitindo assim rastrear requisitos tanto funcionais e não-funcionais como também requisitos de domínio/negócio.

O LAL é meio de rastreabilidade reversa das origens dos RNFs, o que facilita o trabalho da gerência de projetos a controlar a participação dos stakeholders que originam os RNF nas negociações de conflitos de requisitos.

A técnica de TORANZO pode ser utilizada em conjunto com o LAL formando uma solução de rastreabilidade de referências cruzadas bastante poderosa, combinando-se o hipertexto do LAL com as matrizes de rastreabilidade da técnica de TORANZO.

A linguagem OCL pode servir de ponte de integração com o BPMN, tendo potencial para demonstração da complexidade de processos de negócio, como algumas pesquisas recentes mostram (PORCIÚNCULO, 2010). Como já demonstrado em Cysneiros (2001), pode-se demonstrar restrições de negócio ou de processo e restrições técnicas para a implementação dos requisitos funcionais em diagramas UML.

Percebeu-se que o SQFD utilizado em Ramires (2004) pode ser utilizado

juntamente com a metodologia RUP, o que evidencia a possibilidade de uso do SQFD em conjunto com o processo de engenharia de requisitos de Didier (2003).

A análise de compatibilidades abriu caminho para uma reflexão sobre o relacionamento existente entre a proposta de Ramires e as demais propostas estudadas.

Os relacionamentos analisados foram: solução de Ramires com LAL, BPMN, UML, RNF – Framework, técnica de Toranzo.

4.2.1.2.1 Solução de Ramires com LAL

O LAL possui uma característica bastante importante para a solução proposta por Ramires que a de servir de meio para a formação da ontologia à qual as argumentações obedeceriam durante uma instância de processo de negociação. O LAL não se limita à formação de uma ontologia, mas como foi exposto anteriormente no capítulo 2 pode ser usado como instrumento de rastreabilidade, apoiado pelo C&L que é um software para formação do LAL. Essas características que o LAL possui evidenciam a “complementaridade” entre a proposta de Ramires e a estratégia de elicitação e integração de RNF ao requisitos funcionais de Cysneiros. Enquanto a estratégia de Cyneiros se ocupa da elicitação e integração de RNF a requisitos funcionais, a estratégia de Ramires cuida da negociação de conflitos que surgem durante as análises de requisitos.

Uma dificuldade ou incompatibilidade tecnológica do LAL em relação ao MEG esta no fato de que a linguagem de programação utilizada para o desenvolvimento do C&L é o PHP a tecnologia de banco de dados é o MYSQL e as tecnologias utilizadas para desenvolver o MEG são a linguagem de programação VISUAL BASIC, o SGBD ACCESS e planilha EXCEL.

4.2.1.2.2 Solução de Ramires com BPMNRNF

Ainda se poderia desenvolver um módulo no sistema MEG (ou sistema que pudesse ser integrado) para desenvolvimento da visão de processos de negócio que pudesse definir os elementos do BPD (Xavier, 2009) a partir de termos do LAL ou utilizar os BPDs

como ferramentas de comunicação durante o processo de negociação de Ramires (2004). O BPMN tem ganhado popularidade atualmente no mercado por apresentar-se como sendo mais fácil de entender e menos complexo que outras notações e isso é uma vantagem quando se deseja implantar o uso dessa notação como forma de documentação e comunicação visual dos processos de trabalho de uma organização.

A abordagem BPMNRNF (Xavier, 2009) contribui com mais vantagens que podem ser utilizadas para melhorar a solução proposta por Ramires (2004). E traz os benefícios do BPMN e do RNF - Framework, contribuindo, portanto, com a comunicação visual dos processos e restrições já que documenta os processos e restrições. Complementa-se então a comunicação do processo de negociação de Ramires através do uso do BPMN e BPMNRNF.

Um desafio a ser superado com essa integração do BPMNRNF é que não há ferramentas atualmente que suportem a abordagem. O analista terá que utilizar duas ferramentas, uma para BPMN e outra para RNF – Framework. O desenvolvimento de uma ferramenta que suportasse a abordagem seria bastante útil.

4.2.1.2.3 Solução de Ramires com UML

Para a solução proposta por Ramires (o processo de negociação) seria interessante ter uma adaptação da UML para que tenha algumas características da BPMN, que fazem com que ela seja tão popular atualmente no mercado, para que o legado da UML não possa ser perdido e que não se deixe de obter as vantagens da BPMN. Nota-se que há uma tendência atual de comparação entre BPMN E UML:

- a UML é mais robusta;
- porem, a BPMN é mais fácil de ser entendida pelos stakeholders.

A proposta de fusão entre as duas notações é válida pelos seguintes motivos: a estratégia de Cysneiros (2001) tem base na UML, a abordagem BPMNRNF, como o nome já evidencia, tem base no BPMN e traz a integração entre processos de negócio e requisitos não-funcionais. As duas abordagens seriam úteis à solução de Ramires, e não seria interessante

perder o legado deixa por Cysneiros e nem tão pouco o potencial da UML e nem ignorar a abordagem BPMNRNF.

Alguns estudos comparativos (PRIEBE, 2008) evidenciam que essas duas notações tem muito em comum quando se trata de modelagem de processos de negócio e pode-se realizar adaptações na UML para que ela incorpore características do BPMN. Mas essas semelhanças ficam só nesse ponto, pois a UML já possui extensões, tal como a EPBE, que vai além do BPMN.

A EPBE realmente cobre o essencial na modelagem de negócios, estruturando-se em torno de 4 visões: **Visão do Negócio**: uma visão geral do negócio, destacando aspectos estratégicos e táticos (problemas a combater ou oportunidades a aproveitar); **Processos de Negócio**: mostra a dinâmica da organização, inclusive seu relacionamento com entidades externas. **Estrutura do Negócio**: apresenta a estrutura da organização, a divisão de recursos e a carteira de produtos e/ou serviços; **Comportamento do Negócio**: o comportamento individual de cada recurso ou processo no modelo do negócio. (VASCONCELLOS, 2007)

Também seria interessante a combinação da estratégia descrita por Cysneiros (2001) para prover uma rastreabilidade reversa da origem dos requisitos não-funcionais e uma integração entre as visões funcional e não-funcional, que foi conseguido através do uso integrado de UML, OCL (Object Constraint Language – ponto de ligação entre requisitos Funcionais e requisitos Não - Funcionais), RNF - Framework e LAL. Essa rastreabilidade reversa estende a proposta de Ramires facilitando as convocações para reuniões de negociação, o que revela característica de complementaridade de propostas. Tal comodidade promove a participação e comunicação dos participantes. O LAL ajudaria nas argumentações das posições tomadas pelos negociadores diante de um conflito porque que teriam uma ferramenta a mais para relacionar um requisito a outro, entender conflitos, entender posições de outros negociadores e dar respostas mais rápidas às mudanças do projeto.

4.2.1.2.4 Solução de Ramires com RNF – Framework

Nesta pesquisa, verificou-se que é possível a utilização dos grafos do RNF - Framework (CHUNG, 2000) para substituir as matrizes SQFD usadas na solução de Ramires ou usá-las em conjunto com o RNF - Framework.

O processo de negociação de requisitos proposto por Ramires (RAMIRES, 2004)

apresentou o sistema MEG que suporta a negociação para priorização de requisitos. O sistema MEG pode ser customizado ou pode-se desenvolver um outro software com funcionalidades idênticas às do MEG mas acrescidas de mais funcionalidades que suportem os grafos e a análise de conflitos. Poder-se-ia fazer uso de grafos¹⁰ para expor os conflitos entre os requisitos. É válido o esforço de realizar uma adição de uma nova forma de exibição de conflito, em que seriam relacionados e expostos:

- o requisito não-funcional,
- a solução proposta pela equipe (operacionalização do RNF) ,
- as notas dadas pelos negociadores à correlação “RNF v.s solução” e
- a árvore binária de prioridades de negócio¹¹.

Os benefícios a mais, trazidos pelo RNF – Framework ao trabalho de Ramires, são citados na seção 2.4.1 do capítulo dois (2) deste trabalho.

Uma possível dificuldade no momento da transição de matrizes SQFD para grafos é que os grafos não possuem um espaço reservado para atribuição de valores de correlação requisitos/soluções, o significa que os grafos necessitam de adaptações para suportar as atribuições de notas de correlação. Além das adaptações nos grafos, algumas extensões nas ferramentas utilizadas para formar o LAL seriam necessárias para suportar essas características.

4.2.1.2.5 Solução de Ramires com Técnica de rastreabilidade de Toranzo

A solução de Ramires não contempla rastreabilidade ou qualquer outro aspecto de engenharia de requisitos que não seja a negociação e priorização de interesses em decorrência de requisitos conflitantes. Porém seria bastante útil se a solução englobasse a rastreabilidade.

¹⁰No sistema MEG são utilizadas matrizes do SQFD estas matrizes têm como vantagem serem resumidas, o que atribui objetividade em algumas análises, a desvantagem é que só servem para exibir conflitos de interesses entre stakeholders ao passo que os grafos, entre outras vantagens, permite documentar decisões de desenho de grafos, permite exibir as origens dos requisitos etc...

¹¹ A exibição das prioridades de negócio em árvore binária facilita a argumentação para os negociadores de modo que ela seja a mais alinhada possível à governança empresarial e de TIC. A representação visual dos conflitos de requisitos favorece a comunicação entre os negociadores o que também aumenta a confiança entre as partes interessadas já que mais transparência é trazida ao processo de negociação.

A proposta de processo de Engenharia de Requisitos de Didier (2003) fala sobre a técnica de rastreabilidade (TECNICA DE TORANZO), faz referência ao uso do LAL e dos grafos de RNF que são referenciados por Cysneiros (2001). O RNF também é referenciado por Xavier (2009). Esse fato evidencia o alto grau de complementaridade entre as propostas e também mostram uma oportunidade de integração entre essas propostas.

Percebeu-se durante esta pesquisa, como foi dito anteriormente que a rastreabilidade traz mais racionalidade às argumentações dos stakeholders por que alimenta o processo com informações uteis, os que pode ser bastante interessante para as negociações.

4.2.2 Necessidade de adaptações identificadas durante a análise de compatibilidade

Chamaram atenção a possibilidade de realização de adaptações nas correlações UML com BPMN e RNF - Framework com SQFD. Mas também há outros pontos possíveis de adaptar como: adaptações no sistema MEG (para que suporte o LAL), a mesclagem da técnica LAL com a técnica de TORANZO e desenvolvimento de aplicação que suporte essa mesclagem. Outro ponto interessante foi que o trabalho de Cysneiros proporciona a possibilidade de uso em conjunto de sua estratégia com o processo de negociação de Ramires promovendo negociações de qualidade e a formação de uma engenharia de requisitos eficiente.

Adaptações podem ser feitas no RUP e no WRE- Process para que ele se integre com o SQFD e com isso, possa se ter um alinhamento entre RUP, WRE-Process e SQFD. Da mesma forma, poderá-se criar ferramentas, adaptar ou migrar as existente para formar-se um conjunto homogêneo e integrável. Enfim, muitas adaptações poderão ser feitas para tornar as propostas estudadas ainda mais alinhadas umas às outras. Uma análise no apêndice A poderá nortear a identificação de mais pontos a adaptar.

Como desafios às possíveis adaptações nas propostas, identificou-se a heterogenidade de tecnologias a nível de implementação dos softwares de automatização de processos. Entre as tecnologias que foram usadas para implementação dos softwares que apóiam as propostas analisadas, listam PHP, Visual Basic, MySQL, MS-ACCESS, XML, MS-EXCEL, Java, etc. Por conta dessa heterogenidade verifica-se a necessidade de migração dessas tecnologias para um conjunto mais homogêneo, além do desenvolvimento de mais software para automatizar as lacunas das propostas estudadas. A dificuldade fica por conta da necessidade de realizar engenharia reversa dessas ferramentas para se chagar ao modelo conceitual para se elaborar

um outro modelo conceitual que vise integrar essas soluções em uma só. Isso levá a outro ponto a enfrentar que seria o estudo de viabilidade econômica de se realizar essa integração que é algo mais específico ao contexto econômico de uma organização.

5 CONCLUSÕES E TRABALHOS FUTUROS

5.1 Conclusões

De acordo com objetivo desta pesquisa, que era fazer um levantamento e estudo sobre métodos e 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 posteriormente, foi realizado um levantamento e foram encontradas várias compatibilidades entre as propostas estudadas. Sobre a verificação da possibilidade de formulação de uma abordagem metodológica de negociação, esta pesquisa não só confirmou tal possibilidade como encontrou uma solução que já havia sido apresentada por Ramires (2004).

Como uma negociação é algo bastante específico e dependente de um contexto definido, este estudo priorizou pesquisar por propostas que pudessem ser integradas umas com as outras, de forma que se procurou dar mais atenção à análise de compatibilidades entre propostas que se destinassem a resolver problemas deferentes (dentro do contexto de produção e gerência de requisitos) ao invés de realizar análises comparativas entre soluções que resolvessem o mesmo problema. Partiu-se do princípio de que um trabalho que abordasse o tema dessa forma conseguiria explicitar a presença de negociação de requisitos dentro do processo de engenharia de requisitos e também não deixar vago as ligações entre os vários aspectos da engenharia de requisitos que devem ser considerados ao se negociar.

A abordagem do tema desta forma, ou seja, procurando negociação de requisitos em propostas que não tratam diretamente de negociação, deixou claro que o que garante que negociações racionais e produtivas ocorram é o bom funcionamento da infra-estrutura de engenharia de requisitos. Em outras palavras, a análise destas propostas levou a conclusão de que as negociações de requisitos são bem mais suportadas e eficientes quando a engenharia de requisitos é bem feita.

Quanto maior o grau de integração entre as propostas e rastreabilidade entre os artefatos produzidos durante o processo mais agilidade é provida ao negociador. Também maior poder de argumentação conseguiu-se atribuir aos participantes das negociações. Porém em contra partida o trabalho de manutenção dos artefatos aumenta e em consequência a disponibilização de informação aos participantes do processo de desenvolvimento se torna mais dispendiosa, necessitando, assim, de automatização de processos por meio de softwares.

Quanto ao objetivo de verificar ocorrência de compatibilidade com a maioria dos processo de desenvolvimento de softwares atuais, verifica-se que a característica iterativa do trato de requisitos comprova que as negociações se dão de forma também iterativa, o que evidencia a semelhança com processos ágeis e tradicionais como o SCRUM/XP e RUP.

No que diz respeito à intenção de demonstrar a importância do tema Negociação de Requisitos a partir de argumentos, práticas e constatações de autores que estudam o assunto, pode-se dizer que a importância de haver pesquisa sobre tal tema é relevante. Isto se confirma facilmente, pois, como essa pesquisa demonstrou, conflitos podem ser identificados durante as fases iniciais do software. E dentre tais conflitos, pode-se identificar conflitos que não devem ser resolvidos sem que seja levada em consideração a opinião do cliente.

O projeto também tinha como objetivo específicos: procurar por compatibilidades e incompatibilidade entre as propostas selecionada, procurar por pontos carentes de adaptações para torná-las compatíveis entre si, bem como identificar os pontos onde tais adaptações seriam mais fáceis e mas trabalhosas. A esse respeito encontramos compatibilidades entre as propostas, proporcionadas principalmente pela abordagem do LAL e grafos RNFs de (CYSNEIROS, 2001).

Durante esta pesquisa descobriu-se, ao se tentar responder a pergunta “como as propostas estudadas contribuem para prover uma Negociação de qualidade?” que todo negociador tem dois tipos de interesses: na *relação* e na *substância* (Carvalho, 2006). E 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 partes¹² envolvidas nas negociações, porque são fáceis de entender, e são ricos em informações, além de representarem as correlações requisito/ solução (nesse ponto são mais 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, facilitam 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

¹²em Carvalho (2006) há uma diferenciação entre stakeholders (“fantasmas”) e partes envolvidas diretamente nas negociações.

funcional com a possibilidade de se saber como o software irá se comportar antes que ele seja implementado em linguagem de programação, e isso evita perda de tempo e dinheiro.

Pode-se evitar ou diminuir as tensões decorrentes das expectativas de envolvimento das pessoas em situações conflituosas, visto que com a ajuda de tais soluções pode-se chegar a demonstrar aos negociadores o quanto é importante a sua participação. Os grafos possuem um papel muito importante nesse ponto já que podem ser usados para expor problemas que não seriam bem expostos em texto. Expondo-se os problemas, conquista-se a confiança do cliente, o que facilita a sua participação em negociações.

5.2 Trabalhos futuros

Como trabalhos futuros pode-se citar a Integração do BPMNRF (XAVIER, 2009) com o Glossário Léxico (DIDIER, 2003), bem como a utilização em conjunto do SQFD com outras metodologias de desenvolvimento de software como o RUP e suas variações. Também identifica-se como trabalhos futuros o desenvolvimento de ferramenta para produção e controle de Grafos de RNF ou integração com a ferramenta C&L, criação de ferramenta de produção de BPD que suporte a abordagem BPMNRNF (XAVIER, 2009), pesquisa ou desenvolvimento de Ferramenta(s) que suporte(m) a aplicação do WRE-Process (DIDIER, 2003), ferramenta que suporte as funcionalidades do sistema MEG (RAMIRES, 2004) e facilite a sua utilização da estratégia de grafos de RNF proposta por cysneiros (CYSNEIROS, 2001). Também pode-se desenvolver ferramentas para a monitoração dinâmica das negociações e seu desempenho.

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. Pode-se desenvolver um software considere esses fatores.