



## 6º Congresso de Pós-Graduação

### UMA PROPOSTA DE VALIDAÇÃO DE REQUISITOS NÃO FUNCIONAIS UTILIZANDO O NFR-FRAMEWORK.

#### Autor(es)

ANSELMO DE ARAUJO COUTO

#### Orientador(es)

LUIZ EDUARDO GALVÃO MARTINS

#### 1. Introdução

A Engenharia de Requisitos de Software é uma área relativamente nova, tem se desenvolvido muito rapidamente. Em 1993, no primeiro Simpósio Internacional de Requisitos promovido pela *Institute of Electrical and Electronic Engineers (IEEE)*, foram definidas as fronteiras entre a Engenharia de Requisitos e as demais atividades de engenharia de software. A comunidade científica começou a reconhecer a elicitação como algo a ser visto como tema de investigação (LEITE e DOORN 2004).

Para Martins a Engenharia de Requisitos vem crescendo no contexto de desenvolvimento de software, isto se deve porque projetistas de software perceberam que requisitos bem levantados são fundamentais para o sucesso dos projetos (MARTINS 2001).

#### Requisitos

**Sommerville e Sawyer definem requisitos como (SOMMERVILLE e SAWYER 2000):**

*“descrições de como o sistema deve se comportar, com suas propriedades ou atributos”.*

Os autores ainda destacam as descrições dos requisitos nos seguintes aspectos: facilidade no nível do usuário; propriedade geral do sistema; restrição específica sobre o sistema; especificação de um algoritmo particular, que deva ser empregado em cálculos particulares; restrição aplicável ao processo de desenvolvimento do sistema.

Analisando as definições acima, observa-se que requisito é uma declaração sobre o que o sistema proposto irá fazer, documentado e validado pelo cliente para que o problema possa ser resolvido adequadamente e com qualidade. Por esta ótica, há duas categorias de requisitos: aqueles responsáveis pela funcionalidade do sistema e aqueles responsáveis por qualidades que devam estar presentes, tais como desempenho, integridade, disponibilidade e segurança. Os primeiros são denominados requisitos funcionais e os últimos requisitos não-funcionais.

Não há como desconsiderar a importância dos requisitos para o processo de desenvolvimento de software. Entretanto, muitas empresas não têm dado a devida atenção para este fato. Por outro lado, não há mais espaço para que o desenvolvimento continue sendo feito de forma artesanal, ou seja, confiando apenas na experiência e conhecimento técnico do desenvolvedor. Desta forma decisões sobre requisitos são tomadas de forma não sistemática e nem sempre documentada

Além disso, Engenheiros de Requisitos e a maioria das linguagens existentes incidem sobre os requisitos

funcionais, negligenciado ou esquecendo os requisitos não-funcionais (RNFs) durante a concepção do software. Os atributos de qualidade de software (requisitos não-funcionais), por exemplo, são vistos como consequência destas decisões e não como algo que foi pensado. Para mudar este quadro e tratá-los coerentemente, é necessário não apenas métodos e representações, mas também como promover a sua reutilização.

Sendo assim, pesquisas indicam que técnicas baseadas em metas são mais adequadas para requisitos não-funcionais. Pois possibilitam (CHUNG et al. 2000; KAVAKLI e LOUCOPOULOS 2003):

- Examinar a utilidade dos requisitos;
- O grafo SIG (*Softgoal Interdependency Graphs*) permite uma visão vertical desde a estratégia de alto nível até o detalhe;
- Os operadores lógicos E e OU permitem uma melhor tomada de decisão;
- A formalização permite provar a correção e completude dos requisitos não-funcionais.

Desta forma, o *NFR-Framework* proposto por Chung e Mylopoulos é uma abordagem muito utilizada, pois se ajustam a todos os tipos de requisitos não-funcionais, desde as primeiras etapas do processo de desenvolvimento de software (CHUNG et. al. 2000 ; CHUNG e NIXON 1995; MYLOPOULOS et. al. 1992).

### **Visão Geral do *NFR-Framework***

O *NFR-Framework* contribui para que os desenvolvedores tratem os requisitos não-funcionais auxiliando a expressá-los sistematicamente e usá-los para guiar o processo de desenvolvimento de software racionalmente.

O *NFR-Framework* dá uma abordagem orientada a procedimentos para requisitos não-funcionais, onde os RNFs possam ser representados como metas a serem atingidas. Uma característica importante é que as metas contribuirão positivamente ou negativamente, e muitas vezes apenas parcialmente no sentido de uma determinada meta. Portanto, metas de requisitos não-funcionais podem não ser satisfeitas. Assim, o termo satisfação da meta sugere o que é esperado do software dentro de limites aceitáveis para os requisitos não-funcionais. Essa abordagem permite que os aspectos de subjetividade, relatividade e interatividade, inerentes aos requisitos não-funcionais, sejam mais compreensíveis. Desta forma, Chung e Mylopoulos propõem uma noção de *Softgoal* para descrever as metas que precisam ser satisfeitas (MYLOPOULOS et. al. 1992 ; CHUNG et. al. 2000).

O *NFR-Framework* consiste em cinco componentes principais: *Softgoals*, interdependências, procedimento de avaliação, métodos de refinamento e regras de correlação (LAMESWEERDE 2001).

### ***Softgoals***

São as unidades básicas para representar requisitos não-funcionais e são três os tipo de *softgoals*.

*Softgoals* de Requisitos Não-Funcionais - representam requisitos não-funcionais a serem satisfeitos. Elas agem como restrições gerais no sistema.

*Softgoals* de Operacionalização - são técnicas de desenvolvimento que ajudam a satisfazer RNFs, fornecendo mecanismos mais concretos no sistema. Elas fornecem operações, processos, representações de dados, estruturação, limitações e agentes no sistema para satisfazer as necessidades declaradas na s *oftgoals*.

*Softgoals* Declarativas ajudam a justificar decisões. Justifica e explica a lógica do contexto de uma *softgoal* ou a ligação de interdependência.

Figura 1 anexa - Representação Gráfica das *Softgoals*

### **Interdependências**

São interrelações entre os refinamentos de *softgoals* no sentido da satisfação de *softgoals* relacionadas. Existem três tipos de refinamentos:

- Decomposição - Uma *softgoal* é refinada em outra *softgoal* da mesma natureza;
- Operacionalização - O RNF é refinado em *softgoals* operacionalizantes; isto corresponde a uma transição crucial, onde movemos de RNFs (que será alcançado) para o desenvolvimento de técnicas que podem alcançar os RNFs.
- Argumentação - Razões de projeto são anotadas através de argumentação. Elas são rastreadas por refinamento, geralmente envolvendo *softgoals*.

### **Tipos de contribuições:**

Um grupo de metas pode, através do E ou OU, contribuir para suas metas correlatas.

Um único descendente pode contribuir para o seu "pai" nos seguintes níveis: totalmente positiva ("++" ou MAKE), parcialmente positiva ("+" ou HELP), parcialmente negativa (" - " ou HURT), totalmente negativa (" - - " ou BREAK) Este procedimento determina o grau em que um determinado requisito não-funcional é satisfeito por um conjunto de decisões de projeto. Os nós da folha do gráfico de interdependência de *softgoals* são rotulados de acordo com seu grau de satisfação: satisfeito, negado, conflitante, fracamente negado, fracamente satisfeito ou pendente. Os rótulos são propagados até o topo do grafo SIG, em duas etapas. Primeiramente, o impacto individual de cada contribuição em direção ao "pai" é avaliado. Posteriormente, todos os contribuintes para certo "pai" são combinados em um único rótulo. Assim, o processo de avaliação é útil na seleção entre os refinamentos. Desta forma, os desenvolvedores podem usar o procedimento para ver qual o impacto que uma *softgoal* faz em outra *softgoal* .

### **Métodos de refinamento**

Os métodos de refinamento oferecem catálogos de técnicas de desenvolvimento para refinamento de *softgoals* em outras *softgoals*. Tais métodos correspondem a "padrões de refinamento". Durante o processo de desenvolvimento, *softgoals* podem ser refinadas pelo desenvolvedor de uma forma *ad hoc*. Esta tarefa é demorada e torna difícil o conhecimento do domínio compartilhado, prorrogado, adaptados, ou reutilizados. Métodos fornecem refinamentos genéricos que podem ser exemplificados em uma *softgoal*. Eles podem ser retirados de um catálogo que é parte do *NFR-Framework*.

### **Regras de correlação**

Eles fornecem catálogos que permitem inferir possíveis interações, tanto positivas quanto negativas, entre *softgoals*. Os requisitos não-funcionais estabelecidos para um sistema podem estar em conflito ou em harmonia um com o outro. O conflito é representado por correlações negativas, e a harmonia é representada pelas correlações positivas. Assim como os métodos de refinamento, as correlações permitem capturar conhecimento sobre interações genéricas entre *softgoals*, para codificar tal conhecimento, e para compilá-lo em um catálogo. Inicialmente coletado da literatura e da experiência da indústria, as regras de correlação podem ser partilhadas, extendidas, ajustadas e reutilizadas. Detectar correlações negativas pode ser visto como uma troca reveladora, pois pode ser boa para uma *softgoal* ou ao mesmo tempo ruim para outra. Uma questão importante dessa abordagem é que os *softgoals* possuem a propriedade de interagir entre si, em conflito ou sinergia (CHUNG et. al. 2000).

### **Procedimento de avaliação**

## **2. Objetivos**

Engenheiros de Requisitos têm freqüentemente negligenciado ou esquecido os requisitos não-funcionais (RNFs) na concepção do software. A fim de tratá-los coerentemente, é necessário não apenas métodos e representações, mas também como promover a sua reutilização.

O presente trabalho tem como objetivos:

- 1 – Apresentar uma proposta de validação de requisitos não-funcionais utilizando o *NFR-Framework*;
- 2 - Analisar como requisitos não-funcionais são influenciados pelos requisitos funcionais e vice-versa.

## **3. Desenvolvimento**

A proposta metodológica para o desenvolvimento deste trabalho de pesquisa, se dividiu nas seguintes atividades:

- Maior aprofundamento do *NFR-Framework*;
- Estudo da ferramenta *StarUML* (para a modelagem do grafo SIG);
- Definição um roteiro de Validação de Requisitos baseado no *NFR-Framework* ;
- Realização de experimentos aplicando o roteiro de validação proposto;
- Análise dos resultados.

Proposta geral das atividades do roteiro de validação:

- Construir um grafo SIG;
- Validar o grafo SIG;
- Comparar os requisitos não-funcionais antes grafo SIG e pós grafo SIG;
- Verificar a influência dos requisitos não-funcionais sobre os requisitos funcionais;
- Analisar as dependências entre requisitos não-funcionais.

#### 4. Resultado e Discussão

---

Para realização deste trabalho, a fundamentação teórica sobre Engenharia Requisitos, Validação de Requisitos e o *NFR-Framework* foi extremamente importante, levando em conta o domínio de suas características operacionais. Após esta etapa, as especificações de requisitos de dois sistemas serão selecionadas (situação atual do trabalho) e será desenvolvida uma especificação em contra-partida (uma para cada sistema) utilizando o *framework* de requisitos não-funcionais. A proposta de validação será testada e revisada a fim de verificarmos os ganhos de qualidade das especificações de requisitos com a adoção do *NFR-Framework*.

O *NFR-Framework* servirá como um instrumento de melhoria de qualidade no processo de validação de requisitos não-funcionais.

O *NFR-Framework* auxiliará os Engenheiros de Requisitos a efetuar um refinamento dos requisitos não-funcionais do sistema, identificando propriedades do sistema que anteriormente estavam obscuras.

#### 5. Considerações Finais

---

Acreditamos que a utilização do *NRF-Framework* dentro do contexto da validação de requisitos irá contribuir para uma melhoria na qualidade da especificação, modelagem, análise e decisões tomadas ao longo do processo de Engenharia de Requisitos. O *NFR-Framework* torna explícitas as ligações e as soluções entre as *softgoals* e as operacionalizações coerentes em cada projeto. Isto permite uma visualização clara dos objetivos e, por conseguinte, analisar sistematicamente, o quanto o projeto está alinhado com as especificações do usuário. A sua utilização combinada, juntamente com a validação de requisitos, irá ajudar os engenheiros de requisitos a encontrar, com mais facilidade, eventuais falhas no documento inicial de requisitos.

#### Referências Bibliográficas

---

- CHUNG, L. NIXON, B. YU, E. e MULOPOULOS, J. ;“**Non-Functional Requirements in Software Engineering.**” Kluwer Academic Publisher, 2000.
- CHUNG, L. e NIXON, B. ; “**Dealing with Non-Functional Requirements: Three Experimental Studies of a Process-Oriented Approach.**” 17th International Conference on Software Engineering. Abril. 1995.
- KAVAKLI ,E., LOUCOPOULOS, P. ; “**Goal Driven Requirements Engineering: Evaluation of Current Methods**”; EMMSAD , 2003.
- LAMSWEERDE, A. V. ; “**Goal-Oriented Requirements Engineering: A Guided Tour**”*Toronto, Agosto 2001.*, Proceedings RE’01, 5th IEEE International Symposium on Requirements Engineering,
- LEITE, J. C. S. P.; DOORN, J. H. ; “**Perspectives on Software Requirements**”. Kluwer Academic Press, 2004.
- MARTINS, L. E. G. ; “**Uma Metodologia de Elicitação de Requisitos de Software Baseada na Teoria da Atividade**”, Tese de Doutorado, Campinas, SP, agosto de 2001
- MYLOPOULOS, J. , CHUNG, L. NIXON, B. ;“**Representing and Using Non-Functional Requirements: A Process-Oriented Approach**”, IEEE, Canada, 1992.
- SOMMEVILLE, I. e SAWYER, P. ; “**Requirements engineering: A good practice guide**”, Wiley, Chichester, England 2000.



*Softgoals de RNFs*



*Softgoals de Operacionalização*



*Softgoals Declarativas*