

References

SILVA, Reinaldo Antônio da. *Catálogo de Requisitos Não-Funcionais para Sistemas Embarcados: Uma Abordagem Baseada no NFR Framework*. 2019. 104 f. Dissertação (Mestrado em Engenharia de Software) – Universidade Federal de Pernambuco, Recife, 2019.

Lista de Verificação

Itens de Verificação e Referências — NFR Framework (CHUNG et al., 2000)

1. O sistema utiliza o NFR Framework para representar e analisar requisitos não-funcionais?

Trecho (p.30): “O NFR Framework é uma abordagem para representar e analisar Requisitos Não-Funcionais...”

2.4.1 Visão geral

O NFR Framework é uma abordagem para representar e analisar Requisitos Não-Funcionais.

Seu objetivo é ajudar desenvolvedores na implementação de soluções personalizadas, levando em consideração as características do domínio e do sistema em questão. Tais características incluem Requisitos Não-funcionais, Requisitos funcionais, prioridades e carga de trabalho. Esses fatores determinam a escolha de alternativas de desenvolvimento para um

Autor: João Sapiência

2. Os softgoals foram definidos com clareza e sem critérios de satisfação precisos?

Trecho (p.30): “O Framework utiliza o conceito de softgoal: um objetivo que não possui uma clara definição nem critérios de satisfação precisos.”

2.4.1 Visão geral

O NFR Framework é uma abordagem para representar e analisar Requisitos Não-Funcionais. Seu objetivo é ajudar desenvolvedores na implementação de soluções personalizadas, levando em consideração as características do domínio e do sistema em questão. Tais características incluem Requisitos Não-funcionais, Requisitos funcionais, prioridades e carga de trabalho. Esses fatores determinam a escolha de alternativas de desenvolvimento para um determinado sistema (CHUNG et al., 2000).

O Framework utiliza o conceito de softgoal: um objetivo que não possui uma clara definição nem critérios de satisfação precisos. Os softgoals são utilizados para representar Requisitos Não-Funcionais e podem estar inter-relacionados, expressando a influência de um softgoal em outro. O Framework também possui um método de análise qualitativa para decidir os status dos softgoals, dado que outros softgoals relacionados foram ou não satisfeitos (CHUNG et al., 2000).

Autor: João Sapiência

3. As inter-relações entre softgoals estão representadas?

Trecho (p.30): “Os softgoals [...] podem estar inter-relacionados, expressando a influência de um softgoal em outro.”

2.4.1 Visão geral

O NFR *Framework* é uma abordagem para representar e analisar Requisitos Não-Funcionais. Seu objetivo é ajudar desenvolvedores na implementação de soluções personalizadas, levando em consideração as características do domínio e do sistema em questão. Tais características incluem Requisitos Não-funcionais, Requisitos funcionais, prioridades e carga de trabalho. Esses fatores determinam a escolha de alternativas de desenvolvimento para um determinado sistema (CHUNG et al., 2000).

O *Framework* utiliza o conceito de *softgoal*: um objetivo que não possui uma clara definição nem critérios de satisfação precisos. Os *softgoals* são utilizados para representar Requisitos Não-Funcionais e podem estar inter-relacionados, expressando a influência de um softgoal em outro. O *Framework* também possui um método de análise qualitativa para decidir os status dos *softgoals*, dado que outros *softgoals* relacionados foram ou não satisfeitos (CHUNG et al., 2000).

Autor: João Sapiência

4. O modelo inclui análise qualitativa dos softgoals?

Trecho (p.30): “O Framework também possui um método de análise qualitativa para decidir os status dos softgoals.”

2.4.1 Visão geral

O NFR *Framework* é uma abordagem para representar e analisar Requisitos Não-Funcionais. Seu objetivo é ajudar desenvolvedores na implementação de soluções personalizadas, levando em consideração as características do domínio e do sistema em questão. Tais características incluem Requisitos Não-funcionais, Requisitos funcionais, prioridades e carga de trabalho. Esses fatores determinam a escolha de alternativas de desenvolvimento para um determinado sistema (CHUNG et al., 2000).

O *Framework* utiliza o conceito de *softgoal*: um objetivo que não possui uma clara definição nem critérios de satisfação precisos. Os *softgoals* são utilizados para representar Requisitos Não-Funcionais e podem estar inter-relacionados, expressando a influência de um softgoal em outro. O *Framework* também possui um método de análise qualitativa para decidir os status dos *softgoals*, dado que outros *softgoals* relacionados foram ou não satisfeitos (CHUNG et al., 2000).

Autor: João Sapiência

5. Foram utilizados catálogos de requisitos e trade-offs?

Trecho (p.30): “Os catálogos têm a finalidade de organizar o conhecimento sobre Requisitos Não-Funcionais específicos, interdependências e trade-offs.”

2.4.1 Visão geral

O *NFR Framework* é uma abordagem para representar e analisar Requisitos Não-Funcionais. Seu objetivo é ajudar desenvolvedores na implementação de soluções personalizadas, levando em consideração as características do domínio e do sistema em questão. Tais características incluem Requisitos Não-funcionais, Requisitos funcionais, prioridades e carga de trabalho. Esses fatores determinam a escolha de alternativas de desenvolvimento para um determinado sistema (CHUNG et al., 2000).

O *Framework* utiliza o conceito de *softgoal*: um objetivo que não possui uma clara definição nem critérios de satisfação precisos. Os *softgoals* são utilizados para representar Requisitos Não-Funcionais e podem estar inter-relacionados, expressando a influência de um *softgoal* em outro. O *Framework* também possui um método de análise qualitativa para decidir os status dos *softgoals*, dado que outros *softgoals* relacionados foram ou não satisfeitos (CHUNG et al., 2000).

Para lidar com uma variedade de alternativas de desenvolvimento, os desenvolvedores podem consultar os catálogos de requisitos descritos na notação adotada pelo *NFR Framework*.

Os catálogos têm a finalidade de organizar o conhecimento sobre Requisitos Não-Funcionais específicos, interdependências e *trade-offs* (CHUNG et al., 2000).

Autor: João Sapiência

6. O gráfico SIG foi criado para representar softgoals e interdependências?

Trecho (p.31): “Gráfico de interdependência de softgoal conhecido como Softgoal Interdependency Graph (SIG).”

2.4.1 Visão geral

O *NFR Framework* é uma abordagem para representar e analisar Requisitos Não-Funcionais. Seu objetivo é ajudar desenvolvedores na implementação de soluções personalizadas, levando em consideração as características do domínio e do sistema em questão. Tais características incluem Requisitos Não-funcionais, Requisitos funcionais, prioridades e carga de trabalho. Esses fatores determinam a escolha de alternativas de desenvolvimento para um determinado sistema (CHUNG et al., 2000).

O *Framework* utiliza o conceito de *softgoal*: um objetivo que não possui uma clara definição nem critérios de satisfação precisos. Os *softgoals* são utilizados para representar Requisitos Não-Funcionais e podem estar inter-relacionados, expressando a influência de um softgoal em outro. O *Framework* também possui um método de análise qualitativa para decidir os status dos *softgoals*, dado que outros *softgoals* relacionados foram ou não satisfeitos (CHUNG et al., 2000).

Para lidar com uma variedade de alternativas de desenvolvimento, os desenvolvedores podem consultar os catálogos de requisitos descritos na notação adotada pelo *NFR Framework*.

Os catálogos têm a finalidade de organizar o conhecimento sobre Requisitos Não-Funcionais específicos, interdependências e *trade-offs* (CHUNG et al., 2000).

Os softgoals e os seus inter-relacionamentos são representados em um grafo de interdependência de softgoals chamado SIG, descrito na seção a seguir.

Autor: João Sapiência

7. As decisões de desenvolvimento foram registradas graficamente no SIG?

Trecho (p.31): “Os SIGs armazenam um registro completo das decisões de desenvolvimento e da lógica do projeto de forma gráfica e concisa.”

2.4.2 SIG - Softgoal Interdependency Graph

O funcionamento do *NFR framework* pode ser visualizado em termos da construção, elaboração, análise e revisão incremental e interativa de um gráfico de interdependência

31

de *softgoal* conhecido como "*Softgoal Interdependency Graph* (SIG)". Este gráfico registra as considerações do desenvolvedor sobre os *softgoals* e mostra suas interdependências. Os SIGs armazenam um registro completo das decisões de desenvolvimento e da lógica do projeto de forma gráfica e concisa. O registro gráfico das decisões tomadas inclui Requisitos Não-funcionais e suas alternativas, decisões e justificativas associadas às decisões. Para verificar se os requisitos de nível superior são atendidos um procedimento de avaliação pode ser executado (CHUNG et al., 2000).

Autor: João Sapiência

8. Foram identificados os três tipos de softgoals (NFR, Operacionalização e Afirmção)?

Trecho (p.31–32): “Existem três tipos de softgoals: Softgoals NFR, Softgoals de Operacionalização e Softgoals de Afirmção.”

2.4.3 Tipos de Softgoals

Existem três tipos de *softgoals*: *Softgoals NFR*, *Softgoals* de Operacionalização e *Softgoals* de Afirmção. Estes são descritos a seguir:

Autor: João Sapiência

9. Cada softgoal possui tipo e tópico definidos (ex: Confiabilidade [Infusor])?
Trecho (p.32): “A Figura 7 ilustra um Requisito Não-Funcional com tipo ‘Confiabilidade’ e tópico ‘Infusor’.”

Figura 7 – Tipo e tópico de um softgoal



Confiabilidade
[Infusor]

Fonte: (CHUNG et al., 2000)

Autor: Arthur Guilherme

10. Foram aplicados refinamentos top-down entre softgoals?

Trecho (p.33): “Os refinamentos definem um tipo de interdependência que ocorre de cima para baixo (TOP-DOWN).”

2.4.4 Interdependências

As interdependências definem as relações entre os *softgoals*. Os tipos de interdependências utilizadas pelo *framework* são os refinamentos e as contribuições (CHUNG et al., 2000).

2.4.4.1 Refinamentos

Os refinamentos definem um tipo de interdependência que ocorre de cima para baixo (TOP-DOWN), onde um *softgoal* ascendente (pai) produz um ou mais *softgoals* descendentes (filhos) e estes se relacionam com o ascendente. Os tipos de refinamento são: decomposição, operacionalização e afirmação.

As decomposições têm o objetivo de refinar *softgoals* para obter *softgoals* mais especializados e estes possam auxiliar na construção do projeto. Os quatro tipos de decomposições utilizadas pelo *NFR Framework* são descritos a seguir.

Autor: Arthur Guilherme

11. Foram utilizadas decomposições de softgoals (NFR, operacionalização, afirmação e priorização)?

Trecho (p.33): “Os quatro tipos de decomposições utilizadas pelo NFR Framework são [...] Decomposição de Softgoal NFR, de Operacionalização, de Afirmação e Priorização.”

As decomposições têm o objetivo de refinar *softgoals* para obter *softgoals* mais especializados e estes possam auxiliar na construção do projeto. Os quatro tipos de decomposições utilizadas pelo *NFR Framework* são descritos a seguir.

- **Decomposição de Softgoal NFR:** refina ou subdivide um *softgoal NFR* em outros específicos (apresentada na Figura 8-A). Isso pode ajudar a dividir grandes problemas em problemas menores e oferece um aspecto útil para lidar com ambiguidades e prioridades.
- **Decomposição de Operacionalização:** subdivide um *softgoal* de operacionalização em outros *softgoals* de operacionalização mais específicos (apresentada na Figura 8-B). Operacionalizações são úteis para definir uma solução geral e refiná-la em soluções mais específicas.

33

- **Decomposição de Afirmação (*Claims*):** refina um *softgoal* de afirmação em outros *softgoals* de afirmação (apresentada na Figura 8-C). Ela é importante para apoiar ou negar justificativas específicas de projeto.
- **Priorização:** A priorização é um tipo especial de decomposição, onde ocorre o refinamento de um *softgoal* em outro *softgoal* com o mesmo tipo e tópicos, mas com uma prioridade associada. (apresentada na Figura 8-D)

Autor: Arthur Guilherme

12. As contribuições entre softgoals estão representadas e identificadas corretamente?

Trecho (p.34–35): “No NFR framework é possível a utilização de diversos tipos de contribuições que descrevem como a satisfação ou não de um softgoal descendente contribui para a satisfação do ascendente.”

No *NFR framework* é possível a utilização de diversos tipos de contribuições que descrevem como a satisfação ou não de um *softgoal* descendente contribui para a satisfação do *softgoal* ascendente. A seguir apresentamos os tipos de contribuição utilizadas pelo framework (CHUNG et al., 2000):

Autor: Arthur Guilherme

13. Foram utilizadas contribuições positivas e negativas (AND, OR, MAKE, BREAK, HELP, HURT, SOME, UNKNOWN, EQUALS)?

Trecho (p.35–36): “A seguir apresentamos os tipos de contribuição utilizadas pelo framework [...] AND, OR, MAKE(++), BREAK(-), HELP(+), HURT(-), UNKNOWN(?), EQUALS e SOME.”

do *softgoal* ascendente. A seguir apresentamos os tipos de contribuição utilizadas pelo framework (CHUNG et al., 2000):

- **AND**: determina que se os *softgoals* descendentes forem satisfeitos os *softgoals* ascendentes serão satisfeitos.
- **OR**: determina que, se algum *softgoal* descendente for satisfeito, o ascendente será satisfeito.

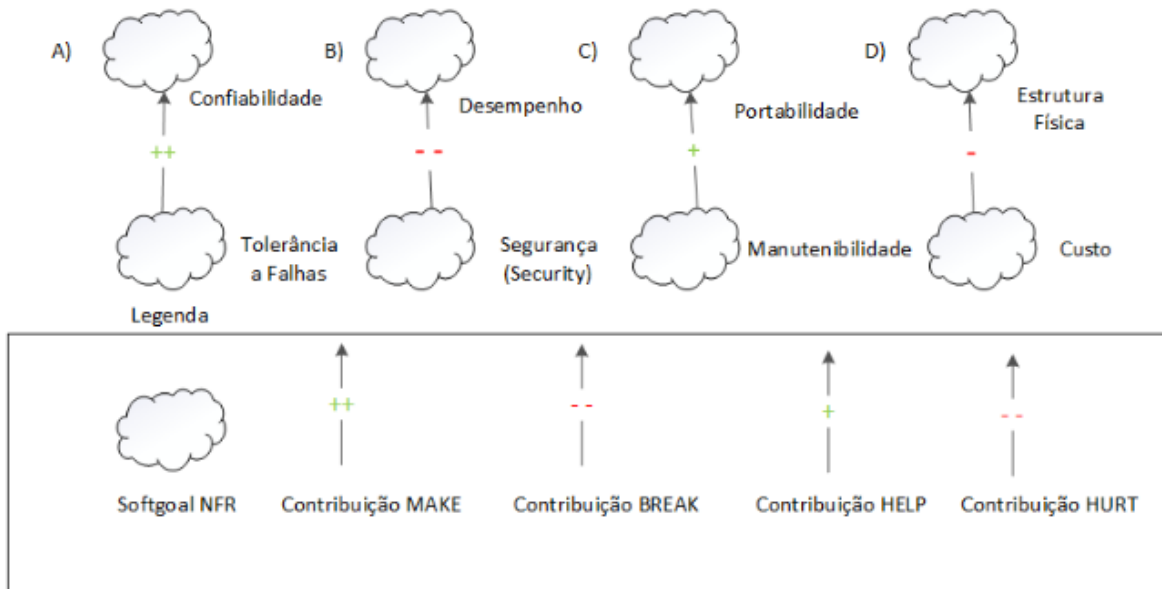
35

- **MAKE(++)**: fornece uma contribuição suficientemente positiva (*MAKE*) entre um *softgoal* descendente e um *softgoal* ascendente que é concebida no nível mais alto de satisfação. Dessa forma, ao utilizarmos *MAKE*, se o *softgoal* descendente for satisfeito o *softgoal* pai também será satisfeito.
- **BREAK(- -)**: fornece uma contribuição suficientemente negativa (*BREAK*) entre um *softgoal* descendente e um *softgoal* ascendente que é concebida no nível mais alto de negação. Portanto, ao utilizar *BREAK*, se o *softgoal* descendente for suficientemente satisfeito o *softgoal* pai será negado, ou seja não será satisfeito.
- **HELP(+)**: fornece uma contribuição parcialmente positiva entre um *softgoal* descendente e um *softgoal* ascendente. Dessa forma ao utilizar *HELP*, se o *softgoal* descendente for parcialmente satisfeito o *softgoal* ascendente será parcialmente satisfeito.
- **HURT(-)**: fornece uma contribuição parcialmente negativa entre um *softgoal* descendente e um *softgoal* ascendente. Dessa forma ao utilizar *HURT*, se o *softgoal* descendente for satisfeito o *softgoal* ascendente será parcialmente negado.
- **UNKNOWN(?)**: fornece uma contribuição desconhecida entre um *softgoal* descendente e um *softgoal* ascendente, podendo ser tanto positiva quanto negativa.
- **EQUALS**: determina que o *softgoal* descendente só será satisfeito se o *softgoal* ascendente for satisfeito e que *softgoal* descendente será negado se o *softgoal* ascendente for negado.

14. Os trade-offs entre requisitos foram analisados (ex: Segurança vs Desempenho)?

Trecho (p.36): “A Figura 12-B apresenta uma contribuição BREAK [...] indicando que soluções de segurança comprometem o desempenho do sistema.”

Figura 12 – Exemplos de contribuições ” MAKE”, ”BREAK”, ”HELP” e ”HURT”



Autor: Arthur Guilherme

15. Foi aplicado o procedimento de avaliação para determinar o grau de satisfação dos RNFs?

Trecho (p.38): “O procedimento de avaliação determina o grau que os requisitos não funcionais são satisfeitos por um conjunto de decisões.”

2.4.5 Procedimento de avaliação

O procedimento de avaliação determina o grau que os requisitos não funcionais são satisfeitos por um conjunto de decisões. Dessa forma, o procedimento de avaliação determina se cada softgoal ou interdependência do SIG foi suficientemente satisfeito. Para isso, são atribuídos rótulos para os softgoals. Os tipos de rótulos utilizados são: satisfeito, fracamente satisfeito, negado, fracamente negado, conflitante, indeterminado. Esses rótulos são ilustrados na figura 14.

Autor: Arthur Guilherme

16. Foram atribuídos rótulos de satisfação aos softgoals?

Trecho (p.38): “Os tipos de rótulos utilizados são: satisfeito, fracamente satisfeito, negado, fracamente negado, conflitante, indeterminado.”

2.4.5 Procedimento de avaliação

O procedimento de avaliação determina o grau que os requisitos não funcionais são satisfeitos por um conjunto de decisões. Dessa forma, o procedimento de avaliação determina se cada softgoal ou interdependência do SIG foi suficientemente satisfeito. Para isso, são atribuídos rótulos para os softgoals. Os tipos de rótulos utilizados são: satisfeito, fracamente satisfeito, negado, fracamente negado, conflitante, indeterminado. Esses rótulos são ilustrados na figura 14.

Autor: Vilmar José

17. O impacto das decisões de baixo nível foi propagado aos níveis superiores do SIG?

Trecho (p.38): “O procedimento de avaliação utiliza esses rótulos para determinar o impacto das decisões nos softgoals que estão em um nível mais alto.”

A análise dos softgoals de nível mais baixo na hierarquia em um SIG se refere a decisões sobre aceitar ou negar alternativas no projeto e essas decisões geram um conjunto inicial de rótulos onde o procedimento de avaliação utiliza esses rótulos para determinar o impacto das decisões nos softgoals que estão em um nível mais alto na hierarquia, isso é realizado até chegar aos softgoals no nível mais alto do SIG.

Autor: Vilmar José

18. O framework contribuiu para rastreabilidade e justificativas das decisões?

Trecho (p.32): “Os softgoals de afirmação fornecem as razões para as decisões de desenvolvimento, facilitando a revisão e o aprimoramento da rastreabilidade.”

- **Softgoals de Afirmação:** permitem que as características do domínio (como prioridades e carga de trabalho) sejam consideradas e devidamente refletidas no processo de tomada de decisão. Eles servem como justificativa para apoiar ou negar a forma como os *softgoals* são priorizados, refinados e os componentes são selecionados. Os *softgoals* de afirmação fornecem as razões para as decisões de desenvolvimento, facilitando a revisão, a justificativa e a mudança do sistema, bem como o aprimoramento da rastreabilidade (CHUNG et al., 2000).

Autor: Vilmar José

19. O NFR Framework foi utilizado para expressar os requisitos do catálogo NFR4ES?

Trecho (p.30): “Este framework é utilizado neste trabalho para representar os Requisitos Não-Funcionais presentes no Catálogo NFR4Es.”

2.4 NFR FRAMEWORK

Este *framework* é utilizado neste trabalho para representar os Requisitos Não-Funcionais presentes no Catálogo NFR4Es, onde estes Requisitos Não-Funcionais serão expressados através de um grafo SIG. O *NFR framework* criado por (CHUNG et al., 2000), foi adotado por propor uma abordagem específica para o tratamento de Requisitos Não-Funcionais e fornecer uma rica representação para expressar esses requisitos, além de suas relações e correlações.

Autor: Vilmar José

20. O modelo final inclui todos os elementos do framework: softgoals, interdependências, contribuições e avaliação?

Trecho (p.39): “Apresentamos uma visão geral sobre este framework e descrevemos seus principais componentes [...] como: softgoals e interdependências, e o procedimento de avaliação dos RNFs.”

Por último, na seção 2.4 abordamos o NFR Framework. Apresentamos uma visão geral sobre este framework e descrevemos seus principais componentes, apresentamos o conceito de grafo SIG e os elementos que compõem este grafo como: softgoals e interdependências. Em seguida, foi descrito o procedimento de avaliação dos RNFs.

Autor: Vilmar José

21. As afirmações foram utilizadas para apoiar ou negar justificativas de projeto?

Trecho (p.34): “As afirmações refinam os softgoals de afirmação [...] Isso é útil para apoiar ou negar motivos específicos de projeto.”

As afirmações refinam os *softgoals* de afirmação em outros *softgoals* de afirmação (*Claim*). Isso é útil para apoiar ou negar motivos específicos de projeto. A figura 10 apresenta a ilustração de possíveis formas de afirmação.

Autor: Vilmar José

22. As contribuições entre softgoals estão corretamente especificadas no SIG?

Trecho (p.34): “No NFR framework é possível a utilização de diversos tipos de contribuições que descrevem como a satisfação ou não de um softgoal descendente contribui para a satisfação do softgoal ascendente.”

No *NFR framework* é possível a utilização de diversos tipos de contribuições que descrevem como a satisfação ou não de um *softgoal* descendente contribui para a satisfação do *softgoal* ascendente. A seguir apresentamos os tipos de contribuição utilizadas pelo framework (CHUNG et al., 2000):

Autor: Vilmar José

23. As contribuições parciais e totais (HELP, HURT, MAKE, BREAK) estão corretamente diferenciadas?

Trecho (p.35–36): “MAKE fornece contribuição suficientemente positiva [...] HELP fornece contribuição parcialmente positiva [...] BREAK e HURT indicam contribuição negativa total ou parcial.”

- **MAKE(++):** fornece uma contribuição suficientemente positiva (*MAKE*) entre um *softgoal* descendente e um *softgoal* ascendente que é concebida no nível mais alto de satisfação. Dessa forma, ao utilizarmos *MAKE*, se o *softgoal* descendente for satisfeito o *softgoal* pai também será satisfeito.
- **BREAK(- -):** fornece uma contribuição suficientemente negativa (*BREAK*) entre um *softgoal* descendente e um *softgoal* ascendente que é concebida no nível mais alto de negação. Portanto, ao utilizar *BREAK*, se o *softgoal* descendente for suficientemente satisfeito o *softgoal* pai será negado, ou seja não será satisfeito.
- **HELP(+):** fornece uma contribuição parcialmente positiva entre um *softgoal* descendente e um *softgoal* ascendente. Dessa forma ao utilizar *HELP*, se o *softgoal* descendente for parcialmente satisfeito o *softgoal* ascendente será parcialmente satisfeito.
- **HURT(-):** fornece uma contribuição parcialmente negativa entre um *softgoal* descendente e um *softgoal* ascendente. Dessa forma ao utilizar *HURT*, se o *softgoal* descendente for satisfeito o *softgoal* ascendente será parcialmente negado.

Autor: João Felipe

24. As relações AND e OR entre softgoals foram utilizadas quando aplicável?

Trecho (p.35): “AND determina que se os softgoals descendentes forem satisfeitos, os ascendentes serão satisfeitos. OR determina que, se algum softgoal descendente for satisfeito, o ascendente será satisfeito.”

- **AND:** determina que se os *softgoals* descendentes forem satisfeitos os *softgoals* ascendentes serão satisfeitos.
- **OR:** determina que, se algum *softgoal* descendente for satisfeito, o ascendente será satisfeito.

Autor: João Felipe

25. Foram aplicadas contribuições SOME, UNKNOWN e EQUALS conforme incerteza das relações?

Trecho (p.37): “A Figura 13-A apresenta uma contribuição Some+, [...] A Figura 13-B apresenta uma contribuição UNKNOWN (?) [...] A Figura 13-C apresenta uma contribuição EQUALS.”

- **UNKNOWN(?):** fornece uma contribuição desconhecida entre um *softgoal* descendente e um *softgoal* ascendente, podendo ser tanto positiva quanto negativa.
- **EQUALS:** determina que o *softgoal* descendente só será satisfeito se o *softgoal* ascendente for satisfeito e que *softgoal* descendente será negado se o *softgoal* ascendente for negado.
- **SOME:** é utilizada quando o sinal da contribuição é conhecido (positivo ou negativo), mas a extensão (parcial ou total) não é. Nesses casos, quando há alguma incerteza em se utilizar *HELP* ou *MAKE* deve-se utilizar o tipo de contribuição *SOME* +. Da mesma forma quando não há certeza em se utilizar *HURT* ou *BREAK* deve-se utilizar *SOME* -.

Autor: João Felipe

26. O procedimento de avaliação foi corretamente aplicado ao final da modelagem?

Trecho (p.38): “O procedimento de avaliação determina o grau que os requisitos não funcionais são satisfeitos por um conjunto de decisões.”

2.4.5 Procedimento de avaliação

O procedimento de avaliação determina o grau que os requisitos não funcionais são satisfeitos por um conjunto de decisões. Dessa forma, o procedimento de avaliação determina se cada softgoal ou interdependência do SIG foi suficientemente satisfeito. Para isso, são atribuídos rótulos para os softgoals. Os tipos de rótulos utilizados são: satisfeito, fracamente satisfeito, negado, fracamente negado, conflitante, indeterminado. Esses rótulos são ilustrados na figura 14.

Autor: João Felipe

27. Foram utilizados os rótulos de satisfação adequados durante a avaliação (satisfeito, negado, etc.)?

Trecho (p.38): “São atribuídos rótulos para os softgoals. Os tipos de rótulos utilizados são: satisfeito, fracamente satisfeito, negado, fracamente negado, conflitante, indeterminado.”

2.4.5 Procedimento de avaliação

O procedimento de avaliação determina o grau que os requisitos não funcionais são satisfeitos por um conjunto de decisões. Dessa forma, o procedimento de avaliação determina se cada softgoal ou interdependência do SIG foi suficientemente satisfeito. Para isso, são atribuídos rótulos para os softgoals. Os tipos de rótulos utilizados são: satisfeito, fracamente satisfeito, negado, fracamente negado, conflitante, indeterminado. Esses rótulos são ilustrados na figura 14.

Autor: João Felipe

28. O impacto das decisões de baixo nível foi corretamente propagado para os softgoals superiores?

Trecho (p.38): “O procedimento de avaliação utiliza esses rótulos para determinar o impacto das decisões nos softgoals que estão em um nível mais alto.”

A análise dos softgoals de nível mais baixo na hierarquia em um SIG se refere a decisões sobre aceitar ou negar alternativas no projeto e essas decisões geram um conjunto inicial de rótulos onde o procedimento de avaliação utiliza esses rótulos para determinar o impacto das decisões nos softgoals que estão em um nível mais alto na hierarquia, isso é realizado até chegar aos softgoals no nível mais alto do SIG.

Autor: João Felipe

29. Há registro gráfico e textual das decisões de refinamento e contribuição?

Trecho (p.31): “O registro gráfico das decisões tomadas inclui Requisitos Não-Funcionais e suas alternativas, decisões e justificativas associadas.”

de *softgoal* conhecido como "*Softgoal Interdependency Graph* (SIG)". Este gráfico registra as considerações do desenvolvedor sobre os *softgoals* e mostra suas interdependências. Os SIGs armazenam um registro completo das decisões de desenvolvimento e da lógica do projeto de forma gráfica e concisa.

O registro gráfico das decisões tomadas inclui Requisitos Não-funcionais e suas alternativas, decisões e justificativas associadas às decisões.

Para verificar se os requisitos de nível superior são atendidos um procedimento de verificação pode ser executado (CHUNG et al., 2000).

Autor: João Felipe