

# NFR Framework

Engenharia de Software

(2007-2008)



Docente: Eng.<sup>a</sup> Isabel Sofia Brito

Discentes: José Janeiro, ei2467

Joaquim Gomes, ei4349





# Âmbito do trabalho

1. Identificação e caracterização dos NFR Frameworks;
2. Contribuições para a Engenharia de Requisitos;
3. Softgoals (SIG – Softgoal Interdependency Graph);
4. Fases básicas para SIG;
5. Vantagens e desvantagens;



## NFR Framework - Caracterização

- Inicialmente proposto em 1992 e desenvolvido por Dr. Lawrence Chung, em 1999;
- *“O NFR Framework é uma abordagem orientada a processos, onde os requisitos não-funcionais são explicitamente representados como metas a serem obtidas” [CHUNG, 1995];*



## NFR Framework – Caracterização (cont.)

- Estão relacionados com o comportamento dum sistema e não com as suas funcionalidades – “*descrevem **como** o sistema faz e não **o que** faz*”[CHUNG];
- São, normalmente, representados por gráficos SIG (Softgoal Interdependency Graph), que descrevem as dependências entre os softgoals e como eles são decompostos;



## NFR Framework – Caracterização (cont.)

- Representam um constrangimento no processo de desenvolvimento:
  - Contraditórios entre si;
  - Difícil modelagem;
  - São críticos;
  - Difícil conciliação. (por vezes é necessário subdividi-los, para chegar à satisfação)



# Tipos de NRF Framework

- Usabilidade;
- Confiabilidade/confidencialidade;
- Acessibilidade;
- Flexibilidade e adaptabilidade, do sistema, para futuras mudanças;
- Escalabilidade, para possíveis incrementos de funcionalidades;
- Performance;
- Eficiência;
- Portabilidade;
- Segurança;
- Robustez;
- Reconfiguráveis.



## NFR Framework - Contribuições

- Estão directamente relacionados com a satisfação dos stakeholders;
- Auxiliam na descoberta de possíveis conflitos entre requisitos;
- Explicitam relacionamentos entre os requisitos que, inicialmente, não eram observados, para obter prioridades entre os requisitos.



## NFR Framework - Contribuições (cont.)

- Decompõe os requisitos, ajudando a entender e quantificar a prioridade de cada requisito em relação às necessidades descritas pelos stakeholders;
- Têm um impacto global na qualidade do sistema;
- Fornecem informações/dados importantes sobre possíveis problemas/conflitos de adaptação e integração do sistema, incluindo restrições na arquitectura (compatibilidade, portabilidade e interoperabilidade) e no próprio projecto (custo e tempo).



# Softgoal

- Torna-se interdependente e, ao mesmo tempo, influente de outros softgoals;
- Representa um objectivo, que não tem definição e/ou critério bem definido, no que respeita à sua satisfação, ou não;
- Diz-se “satisfeitos” quando se aproxima da realização/eficiência do objectivo, ou seja, quando há suficiente evidência positiva e pouca evidência negativa;



## Softgoal - (cont.)

- Existem 3 tipos de representação:
  - Softgoal Refinements (AND/OR); (purificar os objectivos)
  - Softgoal contributions (positive/negative); (possíveis soluções para a satisfação do objectivo)
  - Softgoal operationalizations and claims. (uma possível solução para satisfazer um softgoal e auxilia revisões futuras)



## Softgoal - (cont.)

- Por vezes, há a necessidade de decompor os softgoals, para chegar à satisfação;
- Poderão ser representados numa estrutura gráfica, chamada de SIG (Gráficos de interdependência de softgoals), a fim de se tentar alcançar a satisfação/solução do problema;

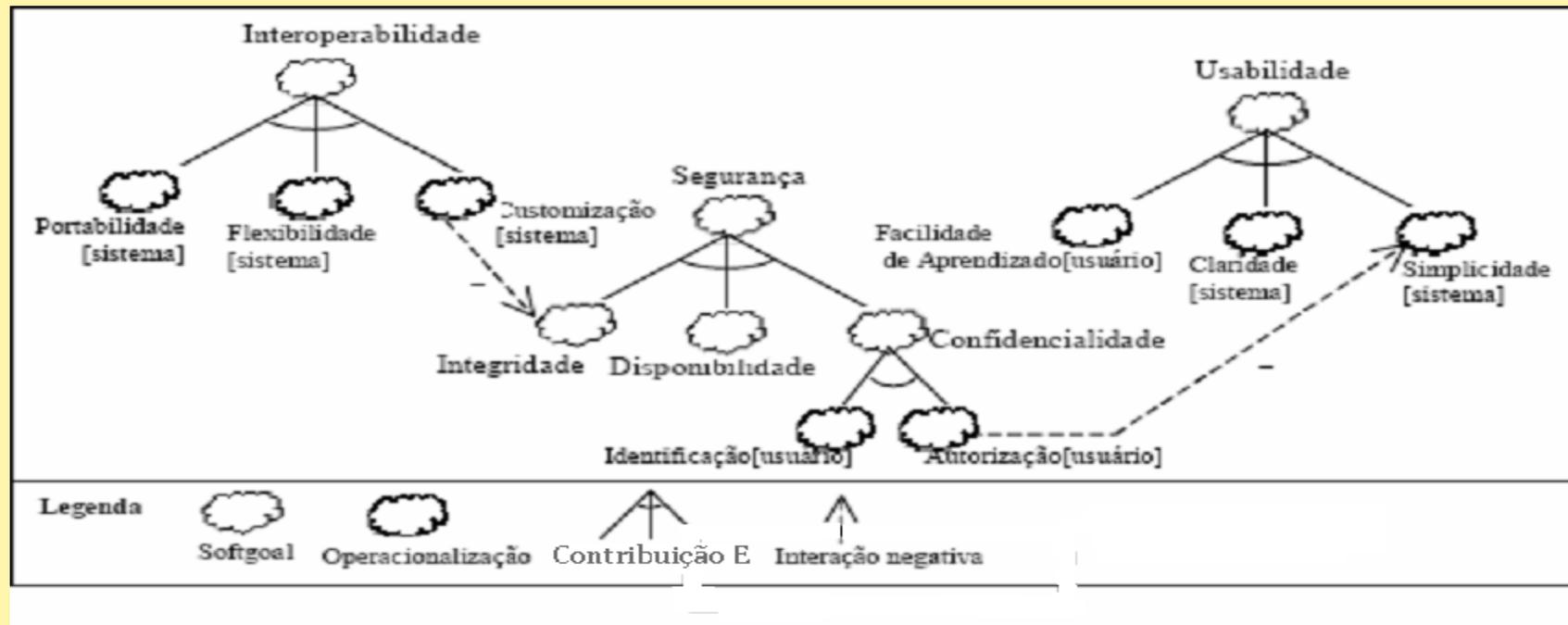


## Fases básicas para SIG

- Desenvolver o objectivo de NFR, bem como os seus refinements (AND/OR), na hierarquia softgoal;
  - **Relação AND** - O objectivo da raiz é satisfatório se todos os seus dependentes forem satisfatórios;
  - **Relação OR** - O objectivo da raiz é satisfatório se, pelo menos, um dos seus dependentes for satisfatórios.

# Fases básicas para SIG - (cont.)

- Decomposição de alguns NFR Framework, utilizando o SIG





## Fases básicas para SIG - (cont.)

- Definir catálogos de conhecimento para originar sistematicamente técnicas de desenvolvimento;
- Identificar correlações entre softgoals;
- Desenvolver críticas ao objectivo;
- Analisar as trade-offs (normalmente perdem uma qualidade para adquirir outra mais adequada) e rationale (relacionar) os softgoals, para esclarecer possíveis ambiguidades;



## Fases básicas para SIG - (cont.)

- Identificar um possível cenário operacional que satisfaça, melhor, as exigências de qualidade dos sistemas (poderá haver mais cenários);
- Relacionar a decisão feita às exigências funcionais, no sistema alvo (não escolher rosa, se as opções são verde ou azul).



## Vantagens

- Ajuda na decomposição dos requisitos;
- Mostra os possíveis conflitos entre requisitos, contribuindo para o processo de tomada de decisão;
- A identificação dos requisitos não funcionais ajuda a desenvolver a aplicação num mais curto espaço de tempo;
- Decompõe os requisitos, ajudando a entender e quantificar a prioridade de cada requisito em relação às necessidades descritas pelos stakeholders;



## Desvantagens

- Devem ser bem analisados, a fim de não comprometer a satisfação dos stakeholders;
- São muito conflituosos entre si; (segurança e performance)
- Nem sempre podem ser totalmente satisfeitos;
- Nem sempre são fáceis identificar;



## Desvantagens - (cont.)

- Os requisitos não funcionais são difíceis de testar, por isso normalmente são avaliados de uma forma subjectiva; (muitas vezes não são levados em conta)
- Difícil equilíbrio – poderá haver casos em que para satisfazer as necessidades, de determinados stakeholders, está-se a comprometer o sistema, por exemplo:
  - Ser complexo de usar (baixa usabilidade);
  - Exigir muita memória (baixa performance);



# Referências

- CHUNG, L., “Non-functional Requirements in Software Engineering”, visitado em 2007, disponível em: <https://www.utdallas.edu/~chung/RE/NFR-18.ppt>;
- CHUNG, L. e NIXON, B., “*Using Non-Functional Requirements to Systematically Support Change*”, 1995;
- Material disponibilizado pela Docente, visitado em 2007, disponível em: [http://www.cs.toronto.edu/~alexei/pub/Lapouchnian-Depth.pdf#search=%22i\\*%20goal%20oriented%20requirements%22](http://www.cs.toronto.edu/~alexei/pub/Lapouchnian-Depth.pdf#search=%22i*%20goal%20oriented%20requirements%22);
- LOUCOPOULOS, P. e KARAKOSTAS, V. “*System Requirements Engineering*”, London, McGraw-Hill, 1995;
- SOMMERVILLE, Ian; “*Software Engineering*”, 7ª Edição, England, Pearson Education Limited, 2004;
- Wikipédia, “*Non-Functional Requirements framework*”, visitado em 2007, disponível em: [http://en.wikipedia.org/wiki/Non-Functional\\_Requirements\\_framework](http://en.wikipedia.org/wiki/Non-Functional_Requirements_framework);
- Wikipédia, “*Interoperabilidade*”, visitado em 2007, disponível em: <http://pt.wikipedia.org/wiki/Interoperabilidade>.