



Requisitos não Funcionais para Sistemas Embarcados - Introdução

Aluno: Reinaldo Antônio da Silva
(ras12@cin.ufpe.br)

Orientador: Jaelson Freire Brelaz de Castro

Coorientador: João Pimentel



UNIVERSIDADE
FEDERAL
DE PERNAMBUCO



Roteiro

- O que são Requisitos?
- Tipos de Requisitos
- Requisitos Não Funcionais
 - A importância dos RNFs
 - Características dos RNFs
 - Exemplos
 - Classificações para os RNFs
 - Abordagens para RNFs
- NFR Framework



Requisitos

- O que são Requisitos?
 - São descrições dos serviços fornecidos pelo sistema e suas restrições operacionais.
- Exemplos:
 - O sistema deve possibilitar o cálculo dos gastos diários, semanais, mensais e anuais com pessoal.
 - O sistema deve emitir relatórios de compras a cada quinze dias.



Tipos de Requisitos

- Requisitos funcionais? São declarações de serviços que o sistema deve fornecer.
 - O software deve possibilitar o cálculo dos gastos diários, semanais, mensais e anuais com pessoal".
 - O software deve emitir relatórios de compras a cada quinze dias"
- Requisitos não funcionais? São restrições sobre os serviços ou as funções fornecidas pelo sistema.
 - Exemplo: a base de dados deve ser protegida para acesso apenas de usuários autorizados".
 - Exemplo: O tempo de resposta do sistema não deve ultrapassar 30 segundos".



Requisitos Funcionais e Não Funcionais

- Requisitos funcionais tem um impacto local
 - As funcionalidades são incorporadas passo a passo no desenvolvimento
 - Afetam apenas a parte do sistema onde as funcionalidades definidas pelo requisito foram implementadas
 - As funcionalidades são implementadas a partir dos requisitos funcionais definidos no documento de requisitos



Requisitos Funcionais e Não Funcionais

- Requisitos não funcionais tem um impacto global
 - A satisfação desses requisitos afeta vários componentes do sistema
 - Não são implementados do mesmo modo que os requisitos funcionais
 - Raramente são considerados durante o processo de desenvolvimento



RNFs na literatura e sua importância

- Não há um consenso sobre RNFs...
- RNFs são importantes para o sucesso do projeto de software.
- Devem ser abordados de forma adequada.
- RNFs Não devem ser Negligenciados
- RNFs são às restrições ou atributos de qualidade sobre o sistema.
- NFRs colocam restrições sobre:
 - o produto que está sendo desenvolvido,
 - o processo de desenvolvimento
 - especificam restrições externas que o produto deve exibir.



Características

- Subjetivos
 - São analisados por diferentes pessoas e podem ter e diferentes interpretações
- Relativos
 - Sua interpretação e relevância depende de cada sistema
- Interativas
 - Podem ter interagir entre si e a satisfação de um requisitos pode inteferir em outro, seja de forma positiva ou negativa.



Exemplos de Requisitos de RNfs

- Performance
- Safety
- Security
- Privacy
- Portability
- Reusability



Classificação dos RNFs

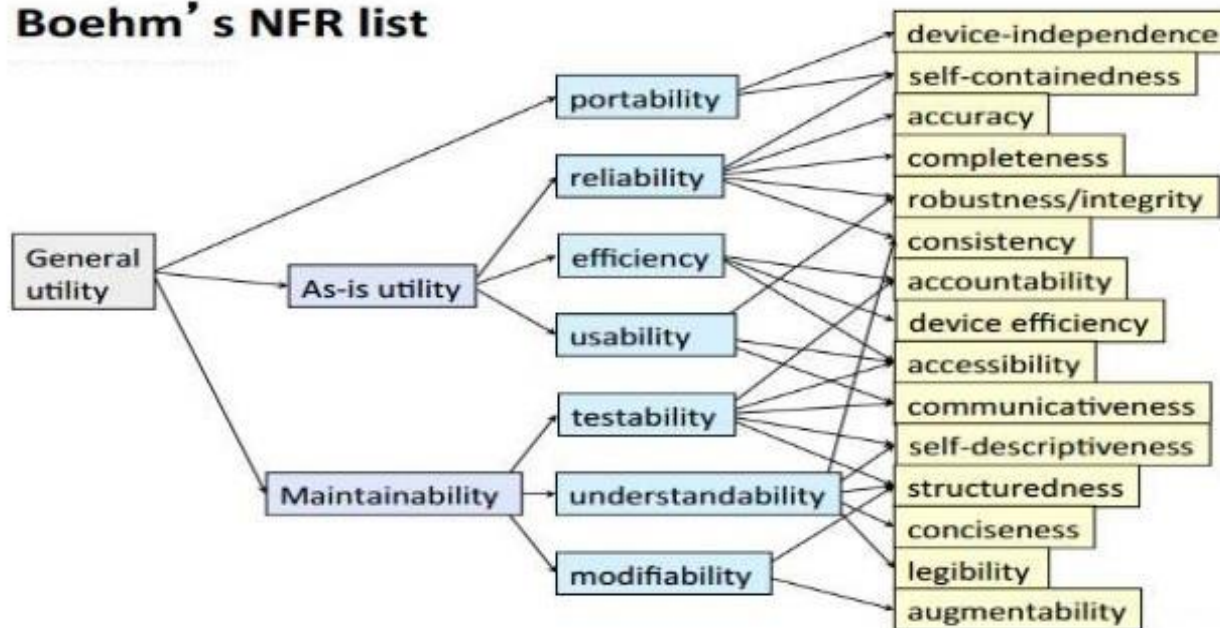
- Na literatura existem varias propostas para classificar requisitos não funcionais
- Não existe uma definição formal ou uma lista completa de requisitos não-funcionais adotada.



Classificação dos RNFs - Boehm

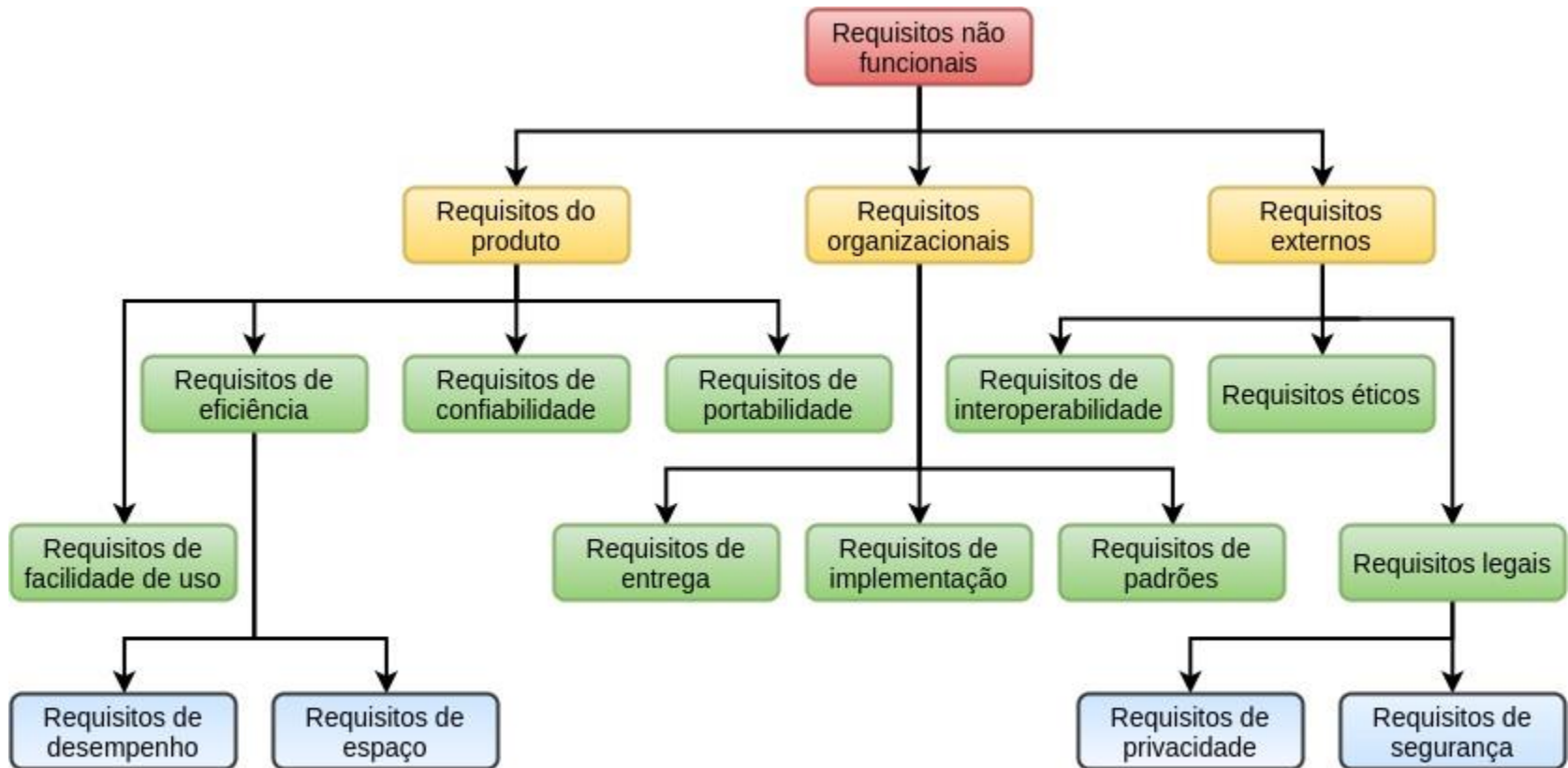
Boehm's NFR list

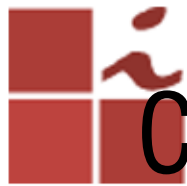
Boehm's NFR list



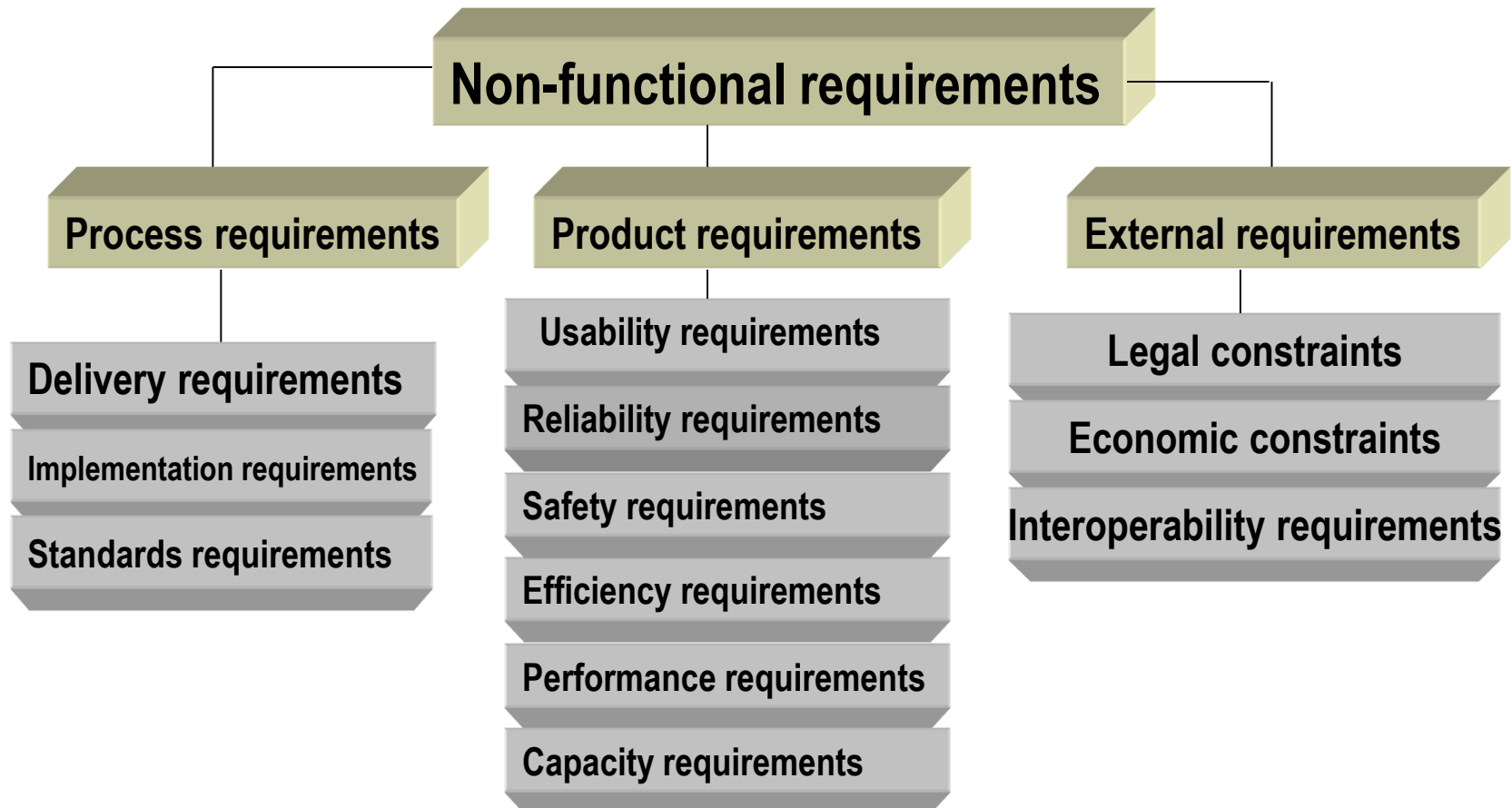


Classificação dos RNFs - Pressman





Classificação dos RNFs - Sommerville





Requisitos de Produto

- São requisitos que especificam as características desejadas que um sistema deve fornecer
 - O sistema deve processar no mínimo 8 transações por segundo (requisito de performance)
 - O sistema deve fornecer um help online (requisito de usabilidade)



Requisitos de Processo

- Restrições relacionadas com o processo de desenvolvimento do sistema
- Normalmente são incluídos em grandes organizações que seguem padrões e práticas pré-estabelecidas
 - O processo de desenvolvimento deve estar de acordo com normas ISO
 - O sistema deve ser desenvolvido usando a linguagem Java



Requisitos de Externos

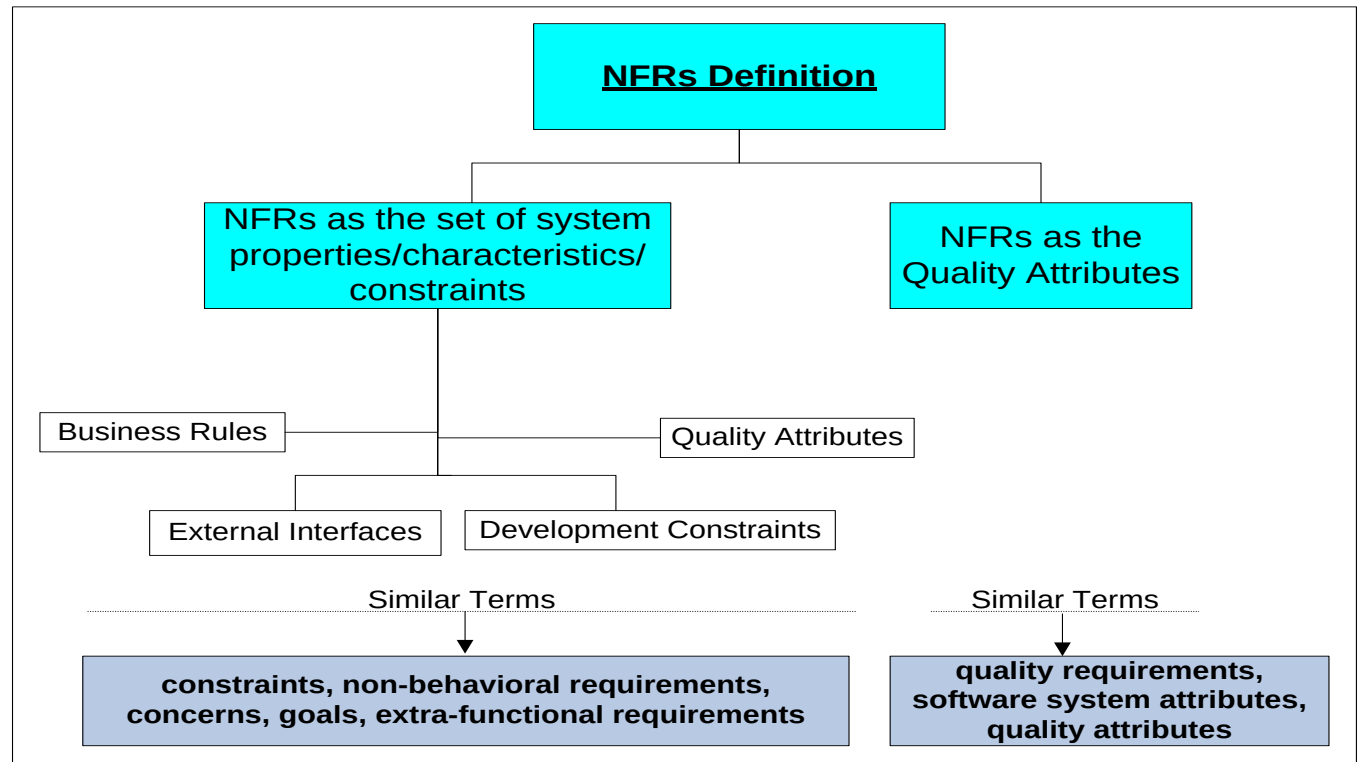
- São derivados do ambiente que o sistema está sendo desenvolvido
- São baseados em informações sobre o domínio de aplicação, considerações organizacionais, restrições de projeto
 - O custo do sistema não deve ultrapassar R\$ 20.000
 - O contrato de manutenção deve incluir o recebimento de novas atualizações do sistema



NFR:Definição

Mairiza & Zowghi [SAC10]

- Pq1: Quantas perspectivas existem na comunidade da engenharia de software quando se considera os NFRs?





NFR:Tipos

Mairiza & Zowghi [SAC10]

- Pq2: Quais os tipos de NFRs ?
 - 252 types of NFRs
 - among these 252 types, 114 types correspond to the NFRs definitions that have been discussed specifically in relation to “the quality”



NFR:Tipos

Mairiza & Zowghi [SAC10]

- | | | | |
|---------------------------------------|----------------------------------|---|--|
| 1. Accessibility/Access Control | 30. Controllability | 60. Legibility | 88. Scalability |
| 2. Accountability | 31. Correctness | 61. Likeability | 89. Security/Control and Security |
| 3. Accuracy | 32. Customizability | 62. Localizability | 90. Self-Descriptiveness |
| 4. Adaptability | 33. Debuggability | 63. Maintainability | 91. Simplicity |
| 5. Additivity | 34. Decomposability | 64. Manageability | 92. Stability |
| 6. Adjustability | 35. Defensibility | 65. Maturity | 93. Standardizability/Standardization/Standard |
| 7. Affordability | 36. Demonstrability | 66. Measurability | 94. Structuredness |
| 8. Agility | 37. Dependability | 67. Mobility | 95. Suitability |
| 9. Analyzability | 38. Distributivity | 68. Modifiability | 96. Supportability |
| 10. Anonymity | 39. Durability | 69. Nomadicity | 97. Survivability |
| 11. Atomicity | 40. Effectiveness | 70. Observability | 98. Susceptibility |
| 12. Attractiveness | 41. Efficiency/Device Efficiency | 71. Operability | 99. Sustainability |
| 13. Auditability | 42. Enhanceability | 72. Performance/Efficiency/Time or Space Bounds | 100. Tailorability |
| 14. Augmentability | 43. Evolvability | 73. Portability | 101. Testability |
| 15. Availability | 44. Expandability | 74. Predictability | 102. Traceability |
| 16. Certainty | 45. Expressiveness | 75. Privacy | 103. Trainability |
| 17. Changeability | 46. Extendability | 76. Provability | 104. Transferability |
| 18. Communicativeness | 47. Extensibility | 77. Quality of Service | 105. Trustability |
| 19. Compatibility | 48. Fault/Failure Tolerance | 78. Readability | 106. Understandability |
| 20. Completeness | 49. Feasibility | 79. Reconfigurability | 107. Uniformity |
| 21. Complexity/Interacting Complexity | 50. Flexibility | 80. Recoverability | 108. Usability |
| 22. Composability | 51. Formality | 81. Reliability | 109. Variability |
| 23. Comprehensibility | 52. Functionality | 82. Repeatability | 110. Verifiability |
| 24. Comprehensiveness | 53. Generality | 83. Replaceability | 111. Versatility |
| 25. Conciseness | 54. Immunity | 84. Replicability | 112. Viability |
| 26. Confidentiality | 55. Installability | 85. Reusability | 113. Visibility |
| 27. Configurability | 56. Integratability | 86. Robustness | 114. Wrappability |
| 28. Conformance | 57. Integrity | 87. Safety | |
| 29. Consistency | 58. Interoperability | | |
| | 59. Learnability | | |



NFR:Tipos

Mairiza & Zowghi [SAC10]

- Among these 114 types,
 - 23 types of NFRs (20.18%) have definition and attributes,
 - 30 types (26.32%) only have definition,
 - and the rest 61 types (53.50%) were introduced without definition or attributes.



NFR:Tipos

Mairiza & Zowghi [SAC10]

- Pq3: Que tipos de NFRs são mais comuns ou considerados na literatura?
- Os cinco mais frequentes tipos de NFRs:
 - performance (88.68%);
 - reliability (67.92%);
 - usability (62.26%);
 - security (60.38%);
 - maintainability (54.72%)



NFR:Tipos

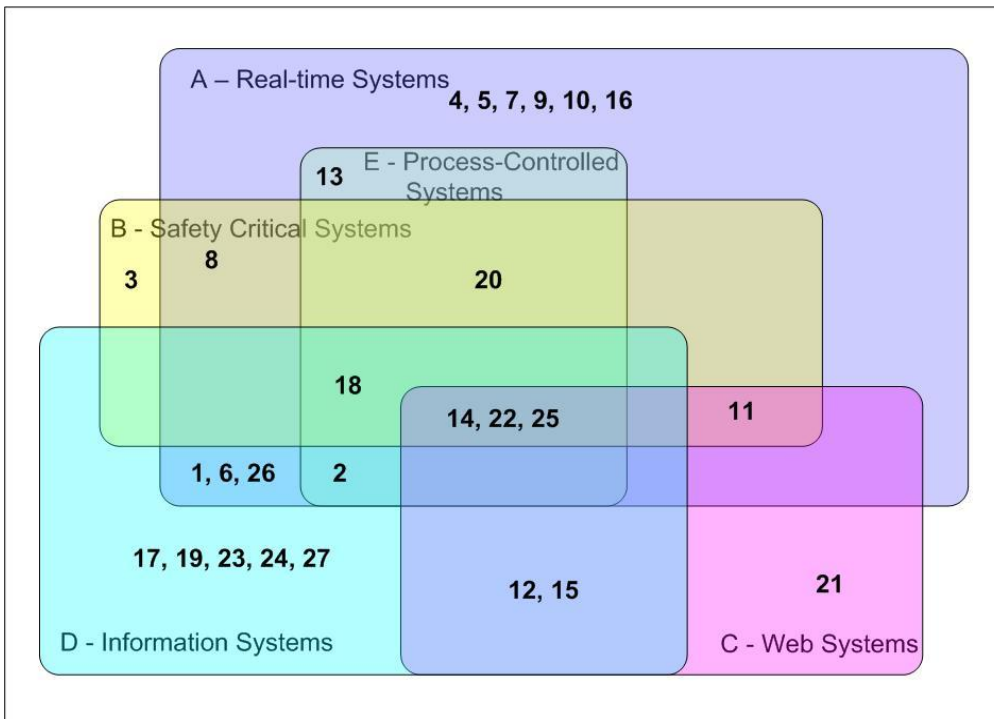
Mairiza & Zowghi [SAC10]

- A investigação também mostrou que alguns tipos de NFRs também são reconhecidos como atributos de outros NFRs
 - Exemplo: integrity, availability, and confidentiality



NFRs, Tipos de Sistemas e Domínios de Aplicação

□ Pq4: Que tipos de NFRs são de interesse em vários tipos de sistemas?



Legend:

1.Accuracy	10.Installability	19.Reusability
2.Availability	11.Integrity	20.Safety
3.Communicativeness	12.Interoperability	21.Scalability
4.Compatibility	13.Maintainability	22.Security
5.Completeness	14.Performance	23.Standardizability
6.Confidentiality	15.Privacy	24.Traceability
7.Conformance	16.Portability	25.Usability
8.Dependability	17.Provability	26.Verifiability
9.Extensibility	18.Reliability	27.Viability



Domínios de Aplicação e NFRs

Mairiza & Zowghi [SAC10]

Application Domain	Relevant NFRs
Banking and Finance	accuracy, confidentiality, performance, security, usability
Education	interoperability, performance, reliability, scalability, security, usability
Energy Resources	availability, performance, reliability, safety, usability
Government and Military	accuracy, confidentiality, performance, privacy, provability, reusability, security, standardizability, usability, verifiability, viability
Insurance	accuracy, confidentiality, integrity, interoperability, security, usability
Medical/Health Care	communicativeness, confidentiality, integrity, performance, privacy, reliability, safety, security, traceability, usability
Telecommunication Services	compatibility, conformance, dependability, installability, maintainability, performance, portability, reliability, usability
Transportation	accuracy, availability, compatibility, completeness, confidentiality, dependability, integrity, performance, safety, security, verifiability



Abordagens dos RNFs

- **Abordagens orientadas a processo**
 - Integra o esforço de descrever e atender NFRs durante o processo de desenvolvimento
- **Abordagens orientadas a produto**
 - Avalia o grau que o produto final atende a determinados NFRs



Abordagens dos RNFs

- **Abordagens orientadas a processo**
 - Ao invés de avaliar a qualidade do produto final, a ênfase é dada em orientar o processo de desenvolvimento do sistema em relação aos NFRs que ele precisa atender
 - As decisões tomadas durante o projeto podem afetar de forma positiva ou negativa RNFs.
 - Essas interdependências servem para explicar o motivo pelo qual o sistema atende ou não a determinado RNF



Abordagens dos RNFs

- **Abordagens orientadas a produto**
 - O sistema é avaliado pelo grau que ele atende a determinado requisito não-funcional
 - Propõe o uso de métricas para medir a qualidade do sistema
 - Existem várias propostas na literatura



Comparativo entre as abordagens

- As abordagens se complementam, não existe uma abordagem melhor que a outra. Devem ser usadas para atender os RNFs dos stakeholders
- Durante o estágio inicial de análise de requisitos é recomendável usar abordagens de processo(requisitos não estão bem definidos)
- Abordagens orientadas a produto são indicadas quando os requisitos estão bem definidos e podem ser especificados em termos de funcionalidades e fatores qualitativos mensuráveis



NFR FRAMEWORK - Princípios

- Proposto por **Chung**, University of Toronto
- Funciona em contraponto as abordadas orientadas a funcionalidade.
- Utiliza requisitos não funcionais para guiar o projeto.
- Representação sistemática e global de NFRs
- Abordagem qualitativa
- Abordagem orientada a processo
- Complementa as abordagens tradicionais orientada a produtos



NFR FRAMEWORK - Princípios

- Fornece uma abordagem orientada a objetivo para lidar com RNFs
 - Ao contrário das estruturas tradicionais os RNFs podem ser raramente satisfeitos de forma completa, o NFR Framework trata da satisfação de softgoals em limites aceitáveis.
- Representa um NFR explicitamente como softgoals
- Coloca os requisitos não funcionais em prioridade para o desenvolvedor.



NFR FRAMEWORK -Visão geral

Etapas ou Atividades Projeto

- Adquirir ou acessar conhecimentos sobre:
 - O domínio particular e o sistema que está sendo desenvolvido
 - Requisitos funcionais para o sistema específico
 - Tipos particulares de NFRs e técnicas de desenvolvimento associadas
- Identificação de NFRs particulares para o domínio
- Decomposição de NFRs
- Identificação de "operacionalizações" (possíveis alternativas de projeto para atender NFRs no sistema alvo)
- lidar com:
 - Ambiguidades
 - trade-offs e prioridades
 - interdependências entre NFRs e operacionalizações
- Selecionar operacionalizações
- Apoiar decisões com racionalidade de projeto
- Avaliar o impacto das decisões.



NFR FRAMEWORK – Visão Geral

Etapas ou Atividades o projeto

- Considerações sobre as atividades do projeto
 - Não devem ser necessariamente realizadas de forma sequencial
 - Podem ser feitas iterações durante o projeto
 - Podem ser realizados refinamentos sobre as operações.



NFR FRAMEWORK – Visão Geral

Componentes do NFR Framework

- Os 5 componentes principais do NFR FRamework
 - Softgoals
 - Interdependências
 - Procedimento de Avaliação
 - Métodos
 - Correlações



NFR FRAMEWORK – Visão Geral

Componentes do NFR Framework

- Softgoals
 - Os softgoal é a unidade básica para representar requisitos não funcionais.
- Tipos de Softgoal
 - Softgoals NFR->representam os RNFs a serem satisfeitos.
 - Softgoals de operacionalização->são técnicas de desenvolvimento que ajudam a satisfazer os softgoal
 - Softgoals de Argumentação->ajudam a justificar decisões



NFR FRAMEWORK – Visão Geral

Componentes do NFR Framework

- Exemplos - Softgoals

Softgoals



NFR softgoal



Operationalizing softgoal



Claim softgoal



NFR FRAMEWORK – Visão Geral

Componentes do NFR Framework

- Interdependências(Refinamentos e contribuições)
 - Registram refinamentos de softgoals pai em softgoals do filho
 - Indicam contribuições do filho para a satisfação dos softgoals dos pais.
 - Podem indicar refinamentos mais gerais entre softgoals e interdependências.



NFR FRAMEWORK – Visão Geral

Componentes do NFR Framework

- Tipos de interdependências – Refinamentos
- Decomposição de NFR
 - Refinam um softgoal em outros softgoals do mesmo tipo.
 - Softgoals NFR, de Operacionalização e de Argumentação
- Decomposição de operacionalização
 - Refinam softgoals de operacionalização em outros softgoals de operacionalização.
 - São úteis para tomar uma solução geral e quebrá-la (operacionalizando-o mais) em mais específicos.



NFR FRAMEWORK – Visão Geral

Componentes do NFR Framework

- Tipos de interdependências – Refinamentos
- Decomposição de Argumentação
 - Refinam os softgoals de argumentação em outros softgoals de argumentação.
 - Úteis para apoiar ou negar motivos específicos de projeto.
- A priorização
 - Refina um softgoal em outro softgoal com o mesmo tipo e tópicos, mas com uma prioridade associada.



NFR FRAMEWORK – Visão Geral

Componentes do NFR Framework

Exemplos:

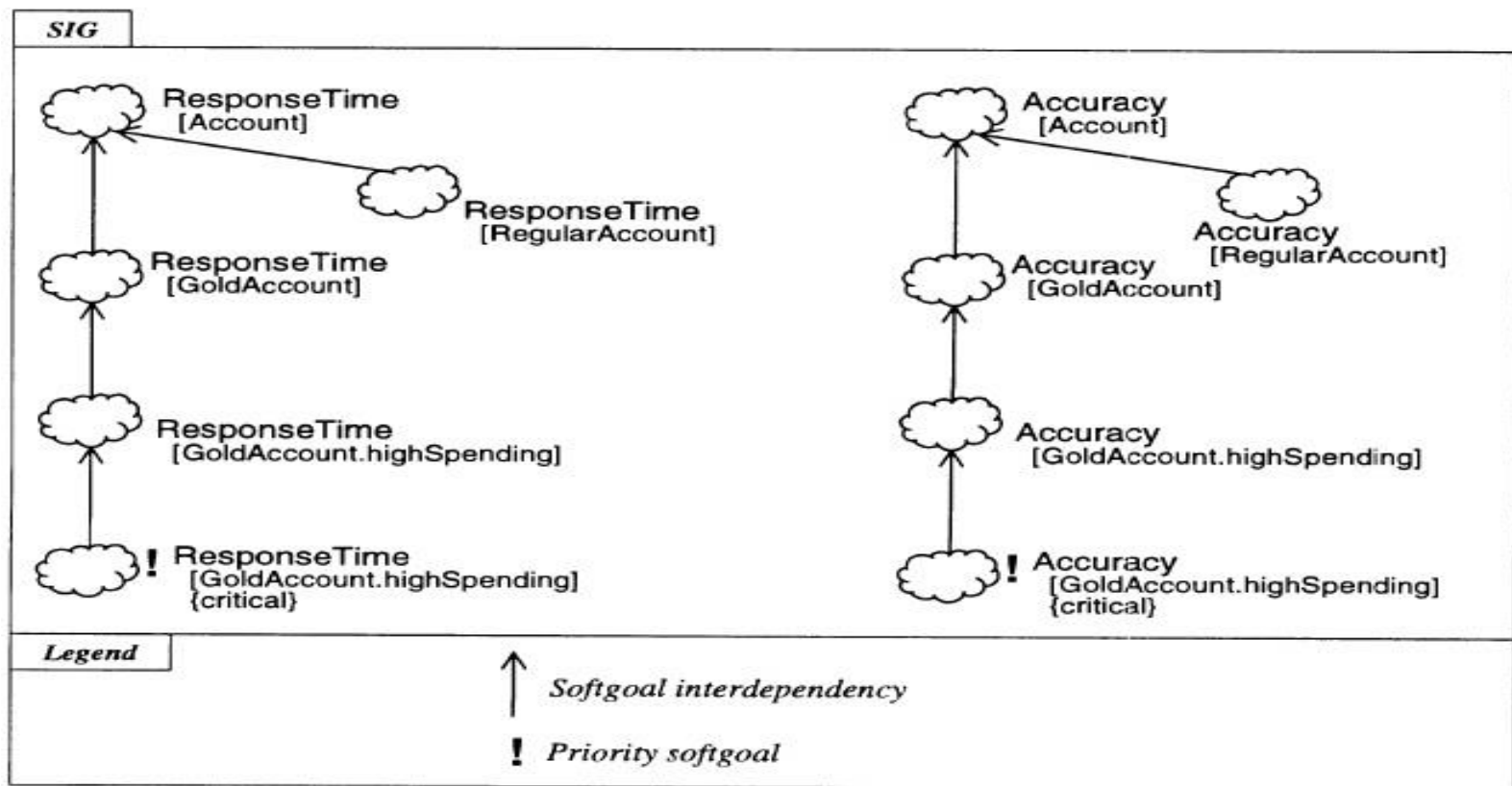


Figure 3.5. Decomposition and prioritization of NFR softgoals.



NFR FRAMEWORK – Visão Geral

Componentes do NFR Framework

- Tipos de interdependências – Refinamentos
- Operacionalizações
 - Refinam softgoals NFR em softgoals de operacionalização.
 - Transição de um softgoal NFR para uma técnicas de desenvolvimento(softgoals de operacionalização) com a finalidade de atingí-lo .
 - Refina um softgoal em outro softgoal com o mesmo tipo e tópicos, mas com uma prioridade associada.
- Argumentação
 - Expressam o raciocínio do projeto. é anotado através da argumentação.



NFR FRAMEWORK – Visão Geral (Softgoal Interdependency Graphs-SIG)

Exemplos:

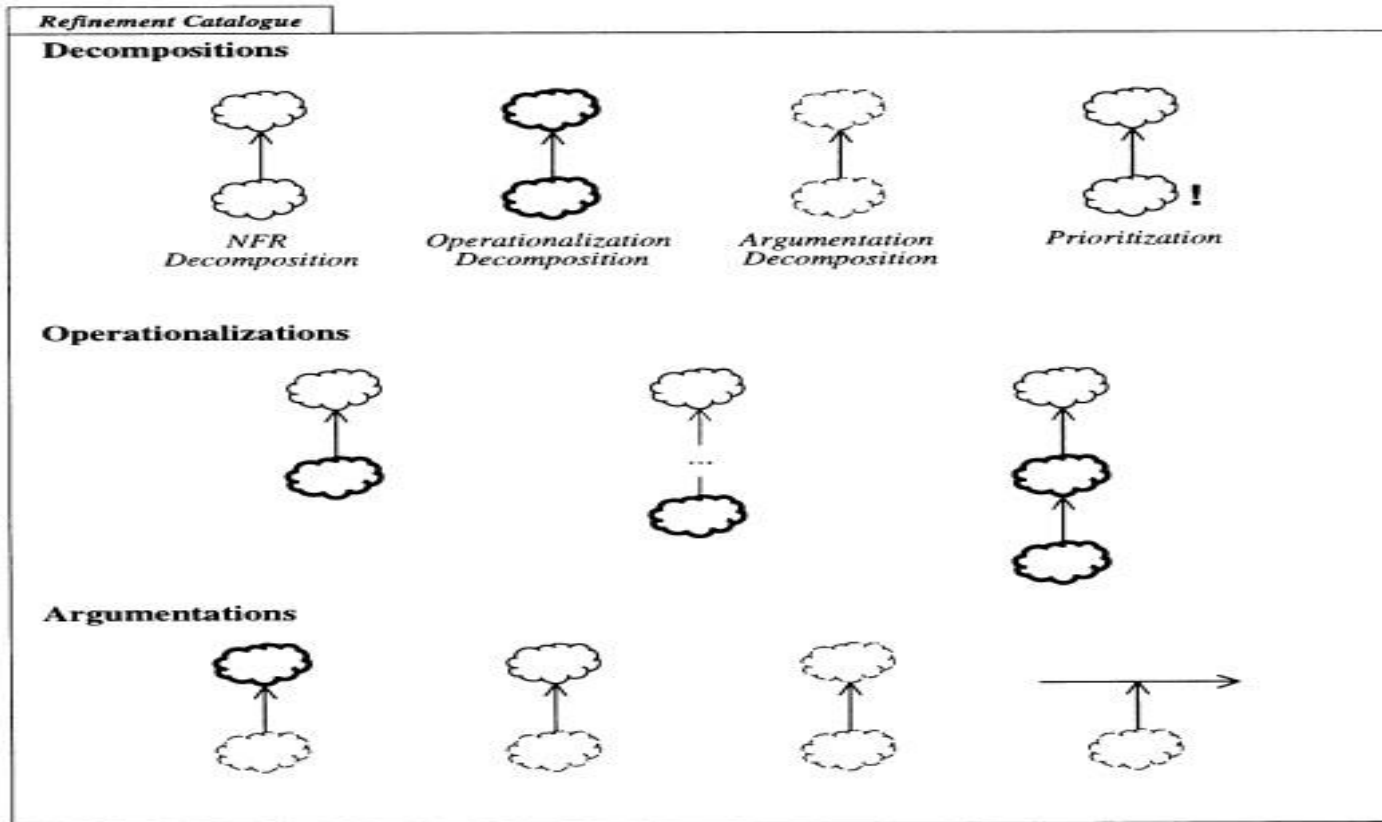


Figure 3.6. Kinds of Refinements.



NFR FRAMEWORK – Visão Geral (Softgoal Interdependency Graphs-SIG)

- Exemplos:

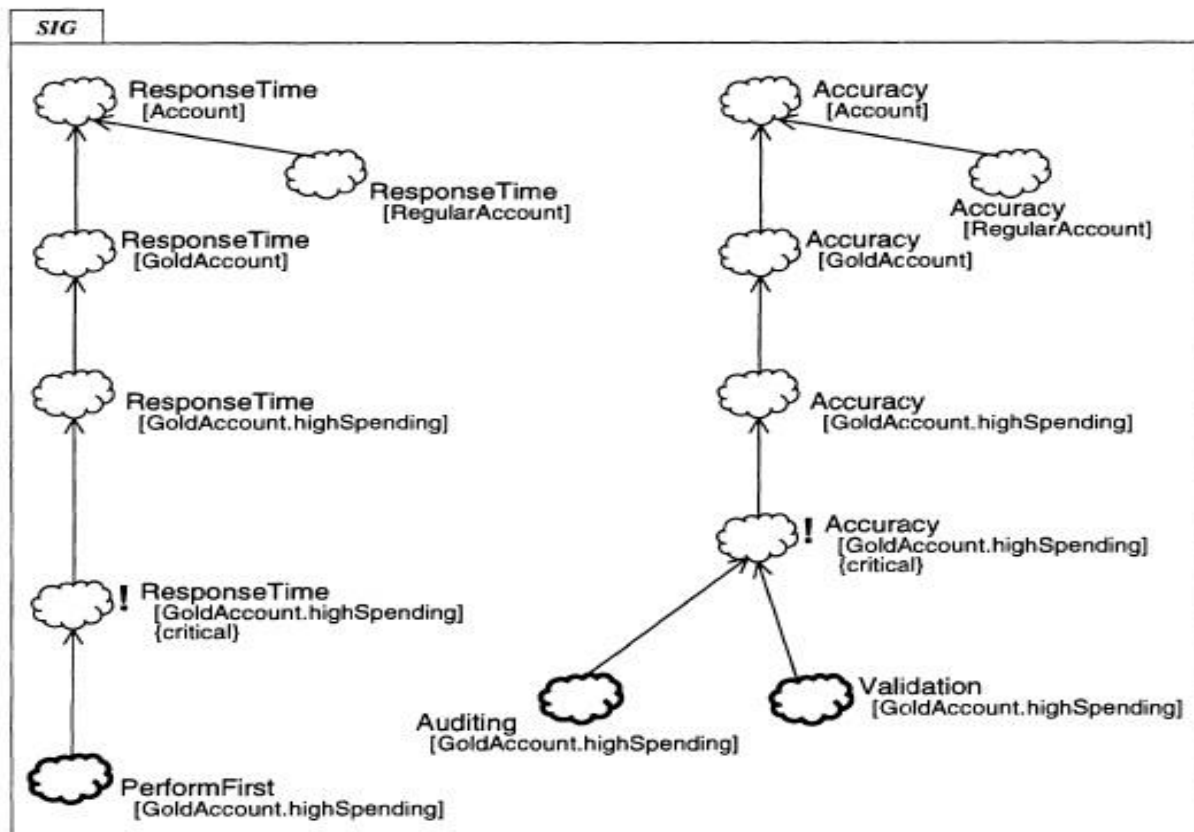


Figure 3.7. Refining NFR softgoals into operationalizing softgoals.



NFR FRAMEWORK – Visão Geral (Softgoal Interdependency Graphs-SIG)

Exemplos:

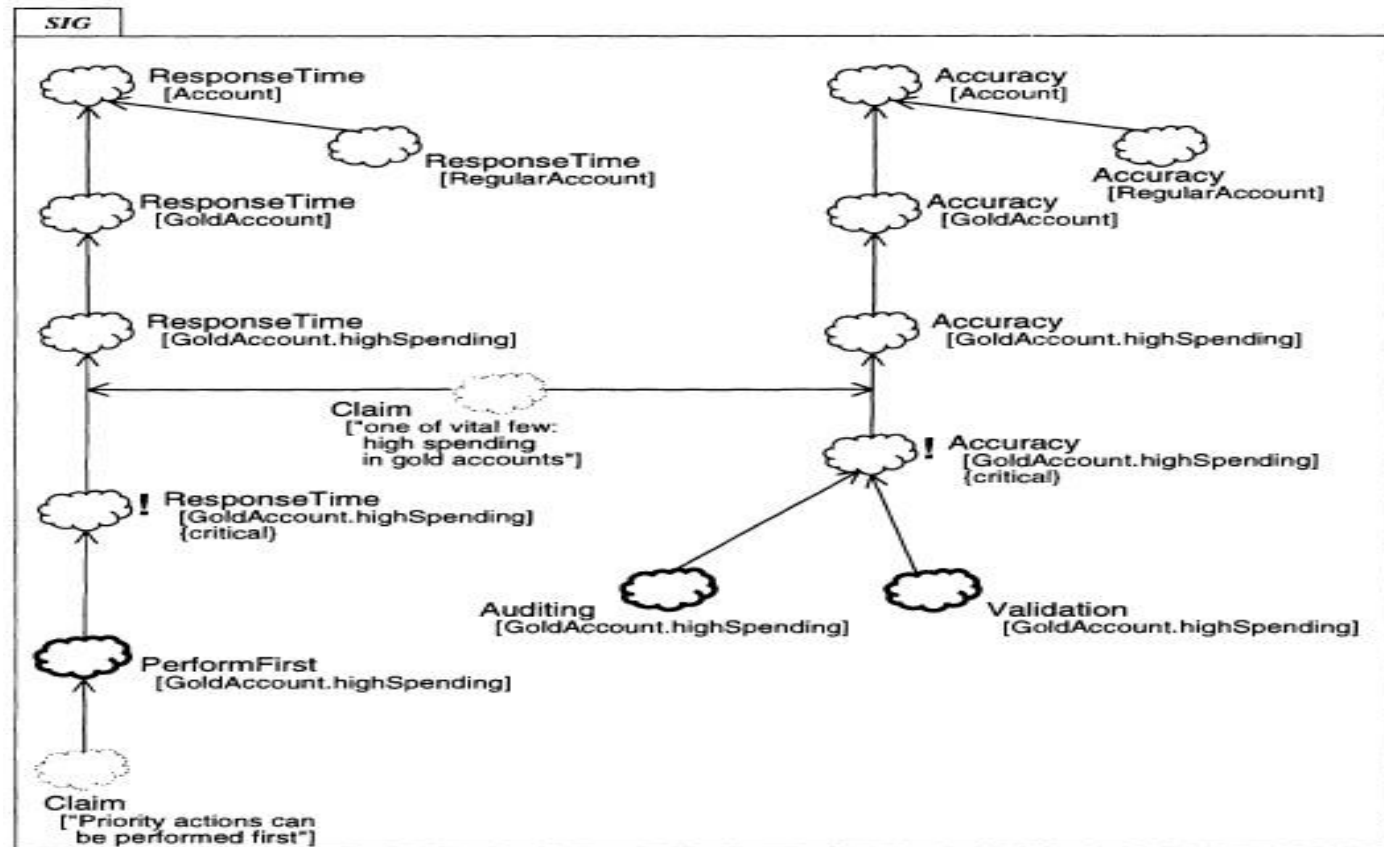


Figure 3.8. Recording design rationale using claim softgoals.



NFR FRAMEWORK – Visão Geral

Componentes do NFR Framework

- Tipos de interdependências – Contribuições
- A medida que o desenvolvimento avança o refinamento dos softgoals "pais" em softgoals "filhos" continua..
- Em alguns refinamentos o filho pode contribuir de forma total ou parcial, e de forma positiva ou negativa, para a satisfação do pai.



NFR FRAMEWORK – Visão Geral

Componentes do NFR Framework

- Tipos de interdependências – Contribuições
- Contribuição AND
 - Tipos de contribuição onde se todos os filhos forem satisfeitos o pai será satisfeito.
- Contribuição OR
 - Tipo de contribuição onde se um dos filhos for satisfeito o pai será satisfeito
- Relacionam um grupo de descendentes a um pai



NFR FRAMEWORK – Visão Geral

Componentes do NFR Framework

- Tipos de interdependências – Contribuições
- Representação (AND e OR)

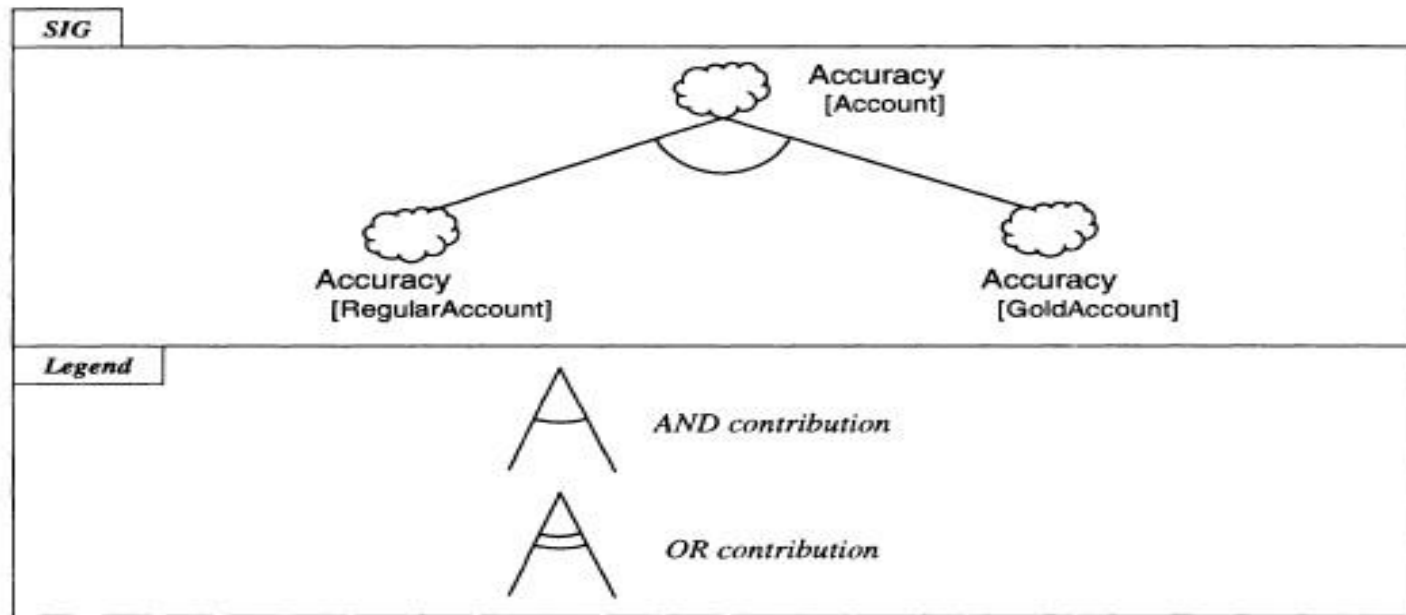


Figure 3.9. A decomposition with an *AND* contribution.



NFR FRAMEWORK – Visão Geral

Componentes do NFR Framework

- Tipos de interdependências – Contribuições
- Contribuições que relacionam um único filho com um único pai:
- MAKE-> fornece suporte positivo total ao pai
 - Se o Filho for satisfeito pai será satisfeito
- BREAK-> fornece suporte negativo total ao pai
 - Se o filho for satisfeito o pai será negado
- HELPS-> fornece suporte positivo parcial ao pai
 - Se o filho for aceito o pai será parcialmente aceito
- HURT-> fornece suporte negativo parcial ao pai
 - Se o filho for negado o pai será parcialmente negado



NFR FRAMEWORK – Visão Geral

Componentes do NFR Framework

- Tipos de interdependências – Contribuições
- Contribuições que relacionam um único filho com um único pai:
- UNKNOWN:
 - faz alguma contribuição, mas de "sinal" desconhecido(positivo ou negativo) e "extensão" (parcial ou suficiente).
- SOME + :fornece alguma contribuição positiva, seja HELP ou MAKE
 - “Some +” = HELP , “Some ++” = MAKE
- SOME - :fornece alguma contribuição negativa, seja HURT ou BREAK
 - “Some -” = HURT , SOME += BREAK



NFR FRAMEWORK – Visão Geral

Tipos de Interdependências – Contribuições

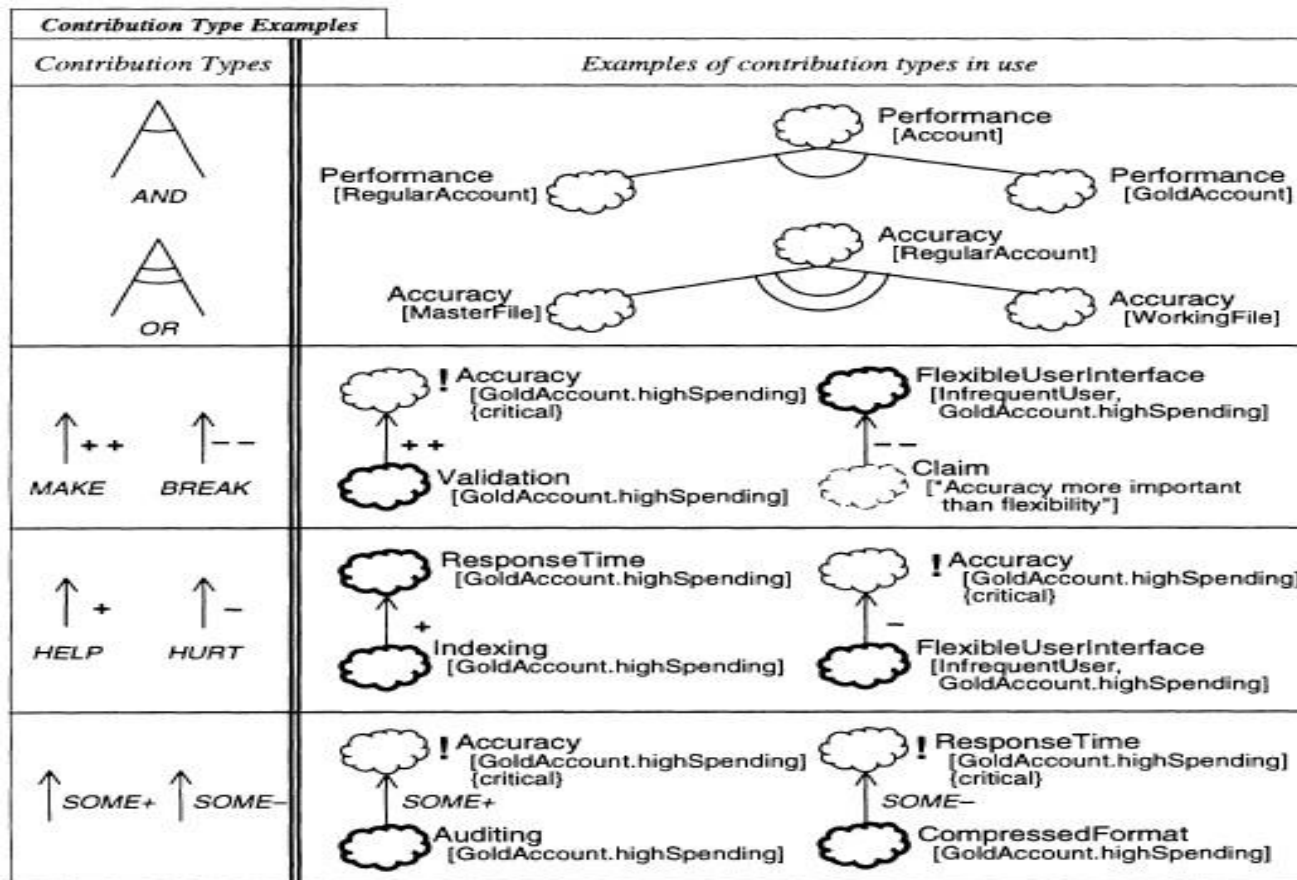


Figure 3.10. Examples of several contribution types.



NFR FRAMEWORK – Visão Geral

Tipos de Interdependências – Contribuições

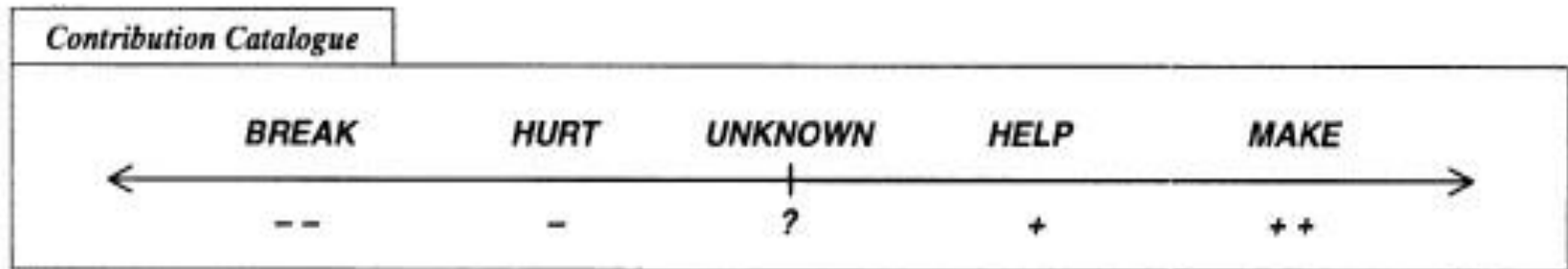


Figure 3.11. Intuitive distinction among contribution types.



NFR FRAMEWORK – Visão Geral

Tipos de Interdependências – Contribuições

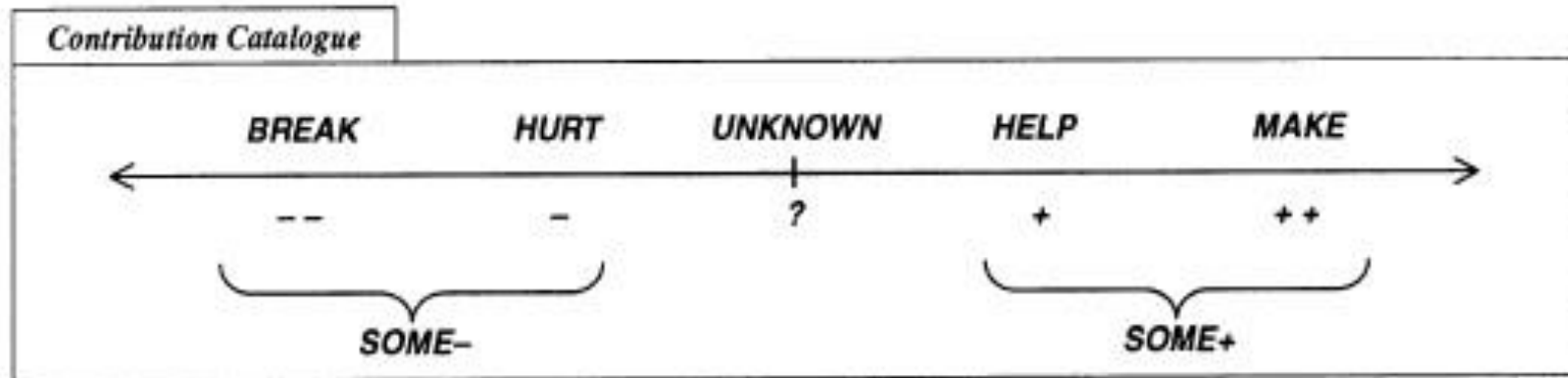


Figure 3.12. Grouping positive and negative contribution types.



NFR FRAMEWORK – Visão Geral

Procedimento de avaliação

- Após a construção do SIG saber se os softgoals foram satisfeitos.
- Essa análise é realizada pelo procedimento de avaliação.
- **O procedimento de avaliação** determina o grau em que os requisitos não funcionais são alcançados por um conjunto de decisões.
- Determina se cada softgoal NFR ou interdependência no SIG está satisfeito.



NFR FRAMEWORK – Visão Geral

Procedimento de avaliação

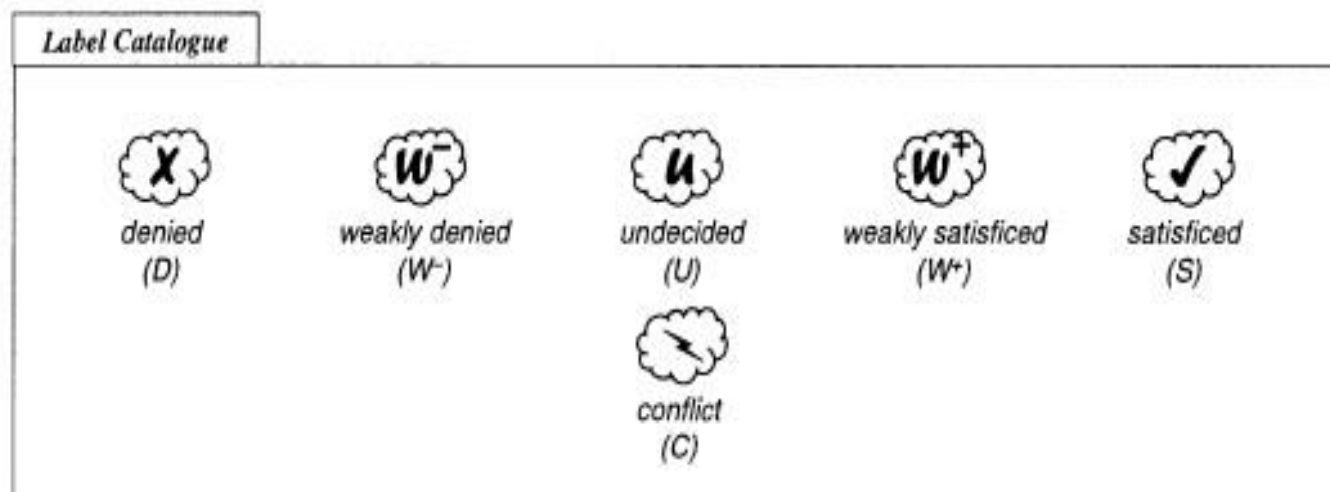


Figure 3.15. Catalogue of label values.



NFR FRAMEWORK

Etapas ou Atividades Projeto

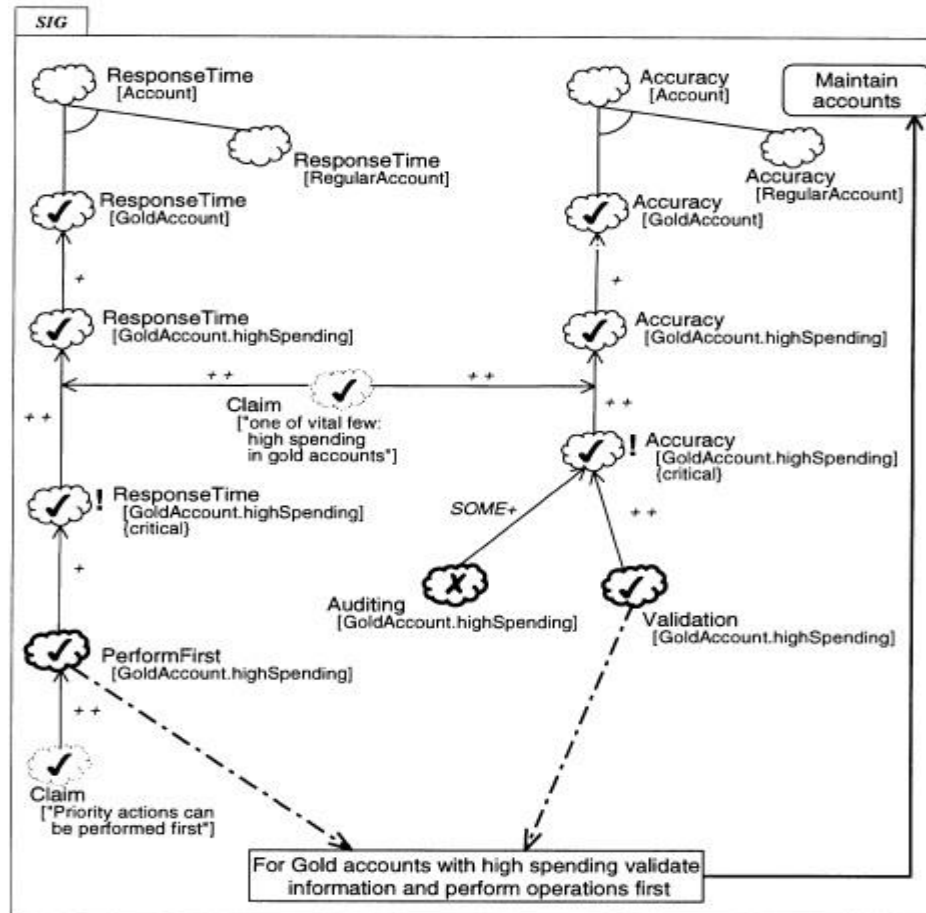


Figure 3.22. Relating functional requirements and the target system to a SIG.



NFR FRAMEWORK – Visão Geral

Catalogos de Requisitos

- NFR Framework permite a possibilidade de armazenar conhecimento sobre experiências anteriores quanto a Requisitos não Funcionais propiciando o reuso.



NFR FRAMEWORK – Visão Geral

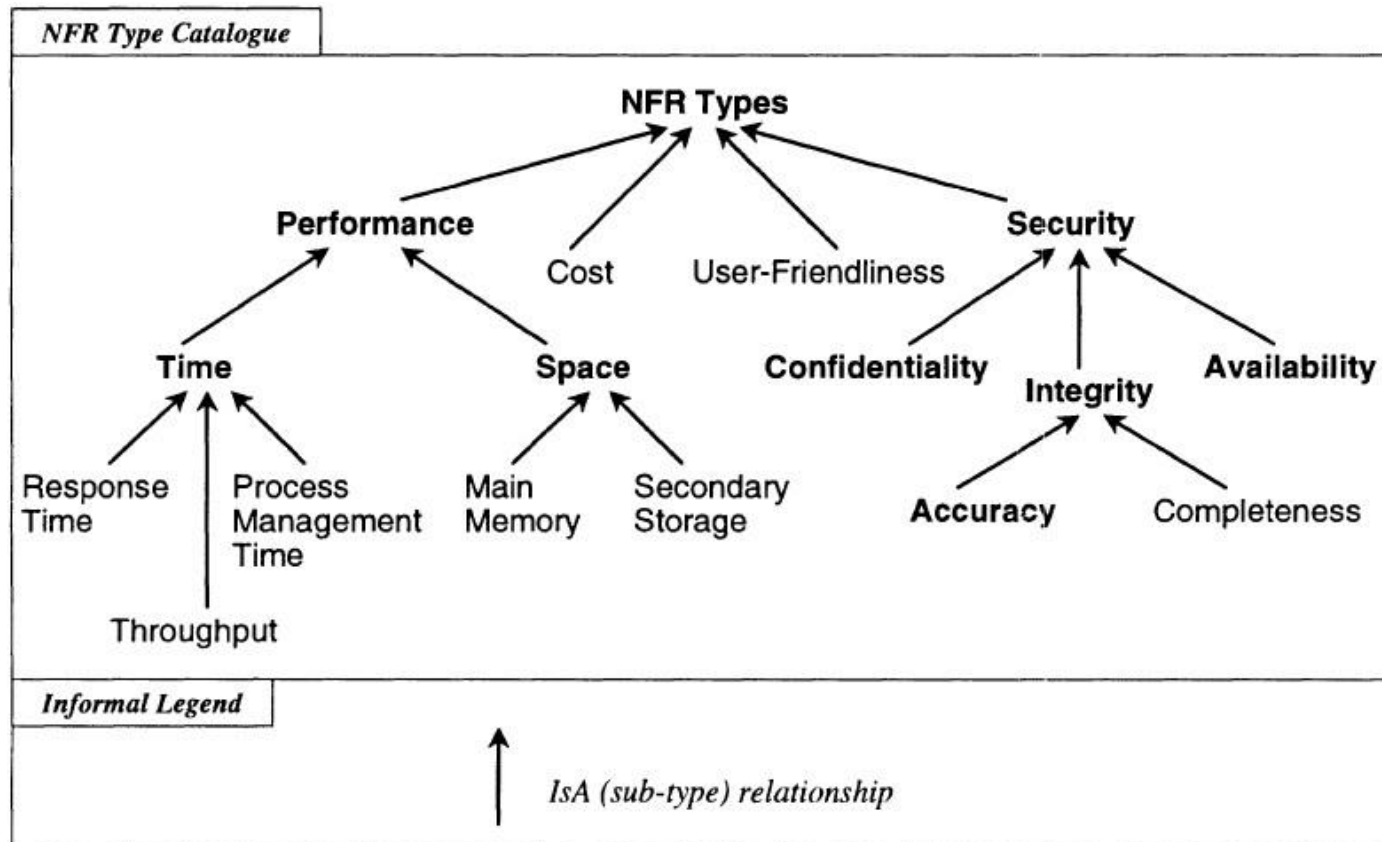
Catalogos de Requisitos

- Existem 3 tipos de catálogo que podem ser usados
 - Catálogo de Requisitos não Funcionais com suas associações de conceitos e terminologia
 - Catálogo de técnicas de desenvolvimento (métodos) para apoiar na descoberta de requisitos
 - Catálogo de interdependências implícitas entre requisitos não Funcionais
- Estes catálogos podem ser criados a partir de diversas fontes (textos, guias de desenvolvimento, livros, especialistas etc.)



NFR FRAMEWORK – Visão Geral

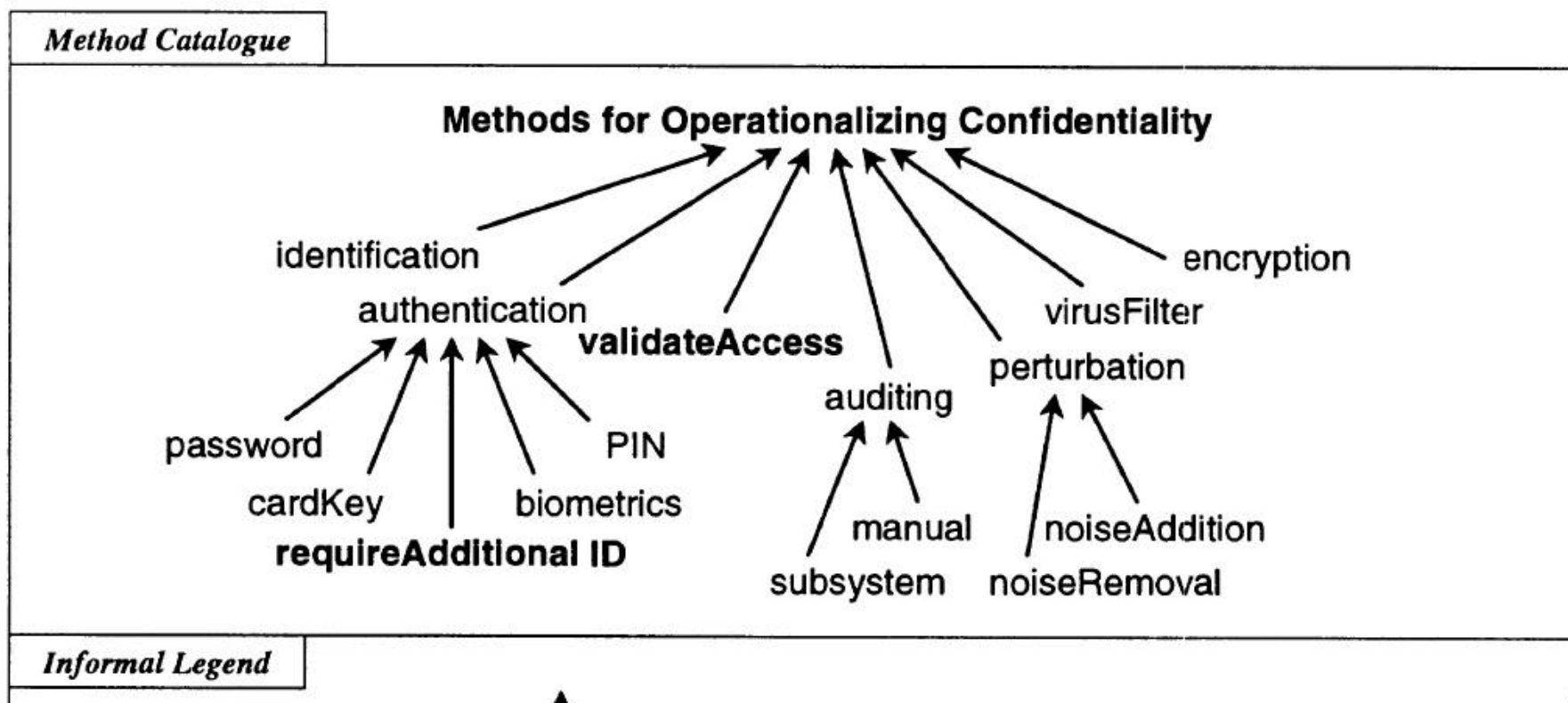
Catalogos de Requisitos





NFR FRAMEWORK – Visão Geral

Catalogos de métodos





NFR FRAMEWORK – Visão Geral

Catalogos de Correlações

<i>Informal Correlation Catalogue</i>					
Impact of offspring <i>Operationalizing Softgoal</i>	upon parent <i>NFR Softgoal</i>				
	Accuracy	Confidentiality	Response Time	Space	User- friendliness
Validation	+	+	-		
Compression			-	+	
Indexing			+		
Authorization		+			
Additional ID		+			-



Próxima apresentação

- Apresentação da abordagem
- Detalhamento da Pesquisa



Referências

- [1] BRAUN, P. et al. Guiding requirements engineering for software-intensive embedded systems in the automotive industry: The REMsES approach. Computer Science - Research and Development, v. 29, n. 1, p. 21–43, 2010.
- [2] Broy M. “Requirements Engineering for Embedded Systems”, Proceedings of FemSys: 1997.
- [3] CYSNEIROS, L.M; YU, E.; LEITE, J.C.S.P., “Cataloguing Non Functional requirements as Softgoal Networks”, In: Proceedings of 11th International Requirements Engineering Conference, pp 13-20, Monterey, California.2003
- [4] CHUNG, L; NIXON, B.; Yu, E; MYLOPOULOS, J. ,2000, Non-Functional Requirements in Software Engineering. Massachusetts.USA. Kluwer Academic Publishers. 2007.
- [5]J.C.S.P. Leite and C. Cappelli, "Software Transparency," Business & Information Systems Engineering: Vol. 2, 2010, Iss. 3, 127-139.
- [6]MILTON DEIVSON ALBUQUERQUE CAVALCANTE. Uma Análise sobre a Engenharia de Requisitos em Sistemas Embarcados.2017. Dissertação de Mestrado do Programa de Pós-Graduação em Ciência da Computação – Universidade Federal de Pernambuco.
- [7]PRESSMAN, R.S, Software Engineering, McGraw-Hill, 6th edition, (2006).



Referências

- [8] ROXANA PORTUGAL, PRISCILA ENGIEL¹, HUGO ROQUE, JULIO CESAR SAMPAIO DO PRADO LEITE. Is There a Demand for Software Transparency ?. Proceeding SBES'17 Proceedings of the 31st Brazilian Symposium on Software Engineering Pages 204-213. 2017.
- [9] TARCÍSIO PEREIRA, DEIVSON ALBUQUERQUE, AÊDA SOUSA, FERNANDA ALENCAR, AND JAELSON CASTRO : Retrospective and Trends in Requirements Engineering for Embedded Systems: A Systematic Literature Review. Anais do WER17 - Workshop em Engenharia de Requisitos, Buenos Aires, Argentina, Maio 22-23, 2017
- [10] KOTONYA, G.; SOMMERVILLE, I. Requirements Engineering: Processes and Techniques. New York, NY, USA: John Wiley & Sons, Inc., 1998.
- [11] OSSADA, J. GERSE: guia de elicitação de requisitos para sistemas embarcados de pequeno e médio porte. 2010. Tese (Doutorado em Ciência da Computação) — Dissertação de Mestrado do Programa de Pós-Graduação em Ciência da Computação— Universidade Metodista de Piracicaba-Piracicaba.
- [12] ZAVE, P. An operational approach to requirements specification for embedded systems. IEEE Transactions on Software Engineering, IEEE Computer Society, Los Alamitos, CA, USA, v. 8, p. 250–269, 1982. ISSN 0098-5589.



Obrigado.