
Requisitos Não-Funcionais

Objetivos

- Introduzir a noção de requisito não-funcional
- Explicar o papel crítico dos requisitos não-funcionais para o sucesso de sistemas de software
- Mostrar abordagens para descrever requisitos não-funcionais

Tipos de Requisitos

□ Funcionais

- Definição das funções que um sistema ou componente do sistema deve fazer
- Ex. O sistema deve permitir a busca de livros por título, autor ou ISBN

□ Não-funcionais

- Relacionados com restrições e aspectos de qualidade
- Ex. O sistema deve ser fácil de usar

Diferenças entre requisitos funcionais e não-funcionais

- ❑ Requisitos funcionais descrevem o que o sistema deve fazer
- ❑ Requisitos não-funcionais fixam restrições sobre como os requisitos funcionais serão implementados
- ❑ Em muitos casos não existe uma clara distinção entre eles

Diferenças entre requisitos funcionais e não-funcionais (cont.)

- Requisitos funcionais tem um efeito localizado
 - Durante o desenvolvimento de sistemas as funcionalidades são incorporadas passo a passo
 - Afetam apenas a parte do sistema onde as funcionalidades definidas pelo requisito foram implementadas
 - No final do processo, as funcionalidades são implementadas a partir dos requisitos funcionais definidos no documento de requisitos

Diferenças entre requisitos funcionais e não-funcionais (cont.)

- Requisitos não-funcionais tem um efeito 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

Principais características dos requisitos não-funcionais (RNF)

- Definem qualidades gerais sobre o sistema
- Relacionados com restrições sobre como os requisitos dos usuários devem ser satisfeitos
 - Ex. restrições de custo, tempo, recursos
- Geralmente são requisitados por gerentes de projeto

A Importância dos requisitos não-funcionais (RNF)

- São críticos para o sucesso de sistemas de software
 - Diretamente relacionados com a satisfação dos usuários



O sistema executa todas as funcionalidades desejadas mas é muito difícil de usar e demora muito para executar operações simples

Exemplos de Requisitos Não-Funcionais

- ❑ Adaptabilidade
- ❑ Confiabilidade
- ❑ Eficiência
- ❑ Flexibilidade
- ❑ Performance
- ❑ Portabilidade
- ❑ Usabilidade

Principais Características dos RNFs

- ❑ **Subjetivos** – eles são interpretados e avaliados por diferentes pessoas que têm diferentes perspectivas e necessidades, assim eles podem ter diferentes significados para cada pessoa
- ❑ **Relativos** – sua interpretação e importância dependem diretamente de cada sistema e sua realização é relativa
- ❑ **Interativos** – eles interagem entre si, assim a realização de um RNF pode interferir positivamente ou negativamente outros requisitos

Problemas para descrever requisitos não-funcionais

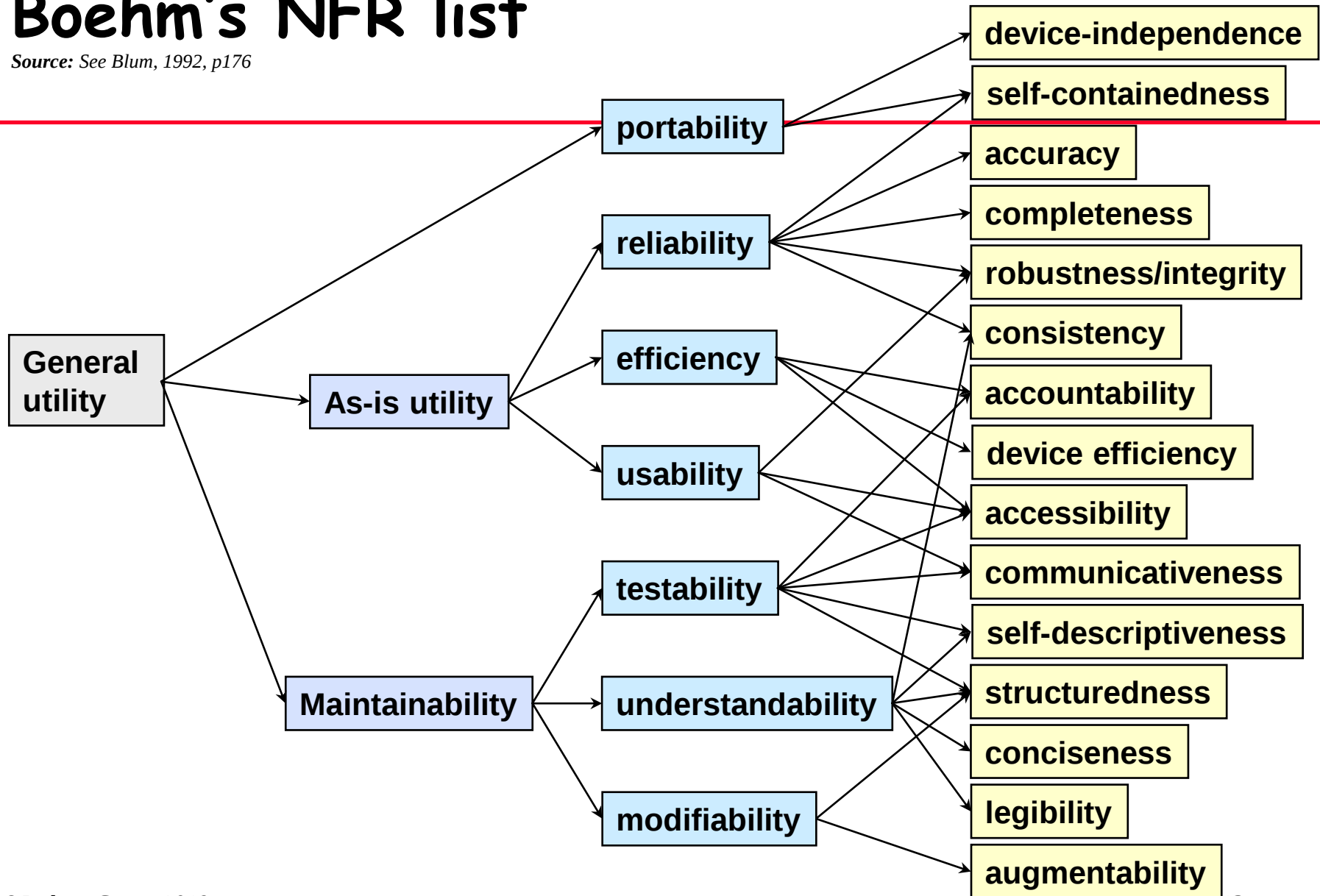
- ❑ Requisitos não-funcionais são difíceis de descrever porém trata-los durante o processo de desenvolvimento pode ser vital para o sucesso de sistemas
- ❑ Como RNFs são críticos, requisitos funcionais podem ser sacrificados para atender as restrições impostas pelos requisitos não-funcionais

Classificação dos requisitos não-funcionais

- Não existe uma definição formal ou uma lista completa de requisitos não-funcionais
- Existem várias propostas para classificar RNFs
 - Lista Boehm
 - Lista MacCall
 - IEEE-Std 830-1998 lista 13 requisitos não-funcionais
 - Sommerville classifica requisitos não-funcionais em 3 categorias

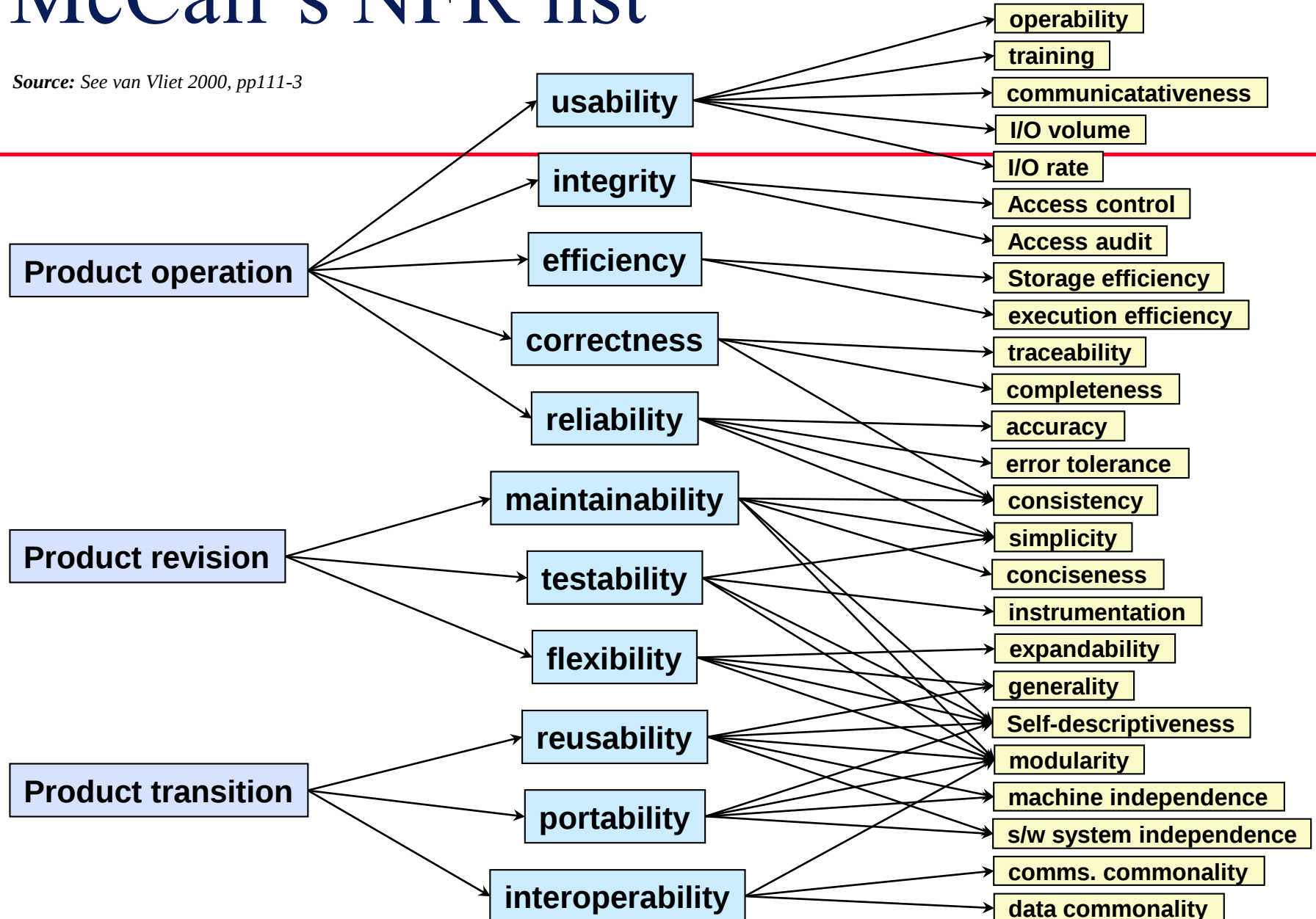
Boehm's NFR list

Source: See Blum, 1992, p176



McCall's NFR list

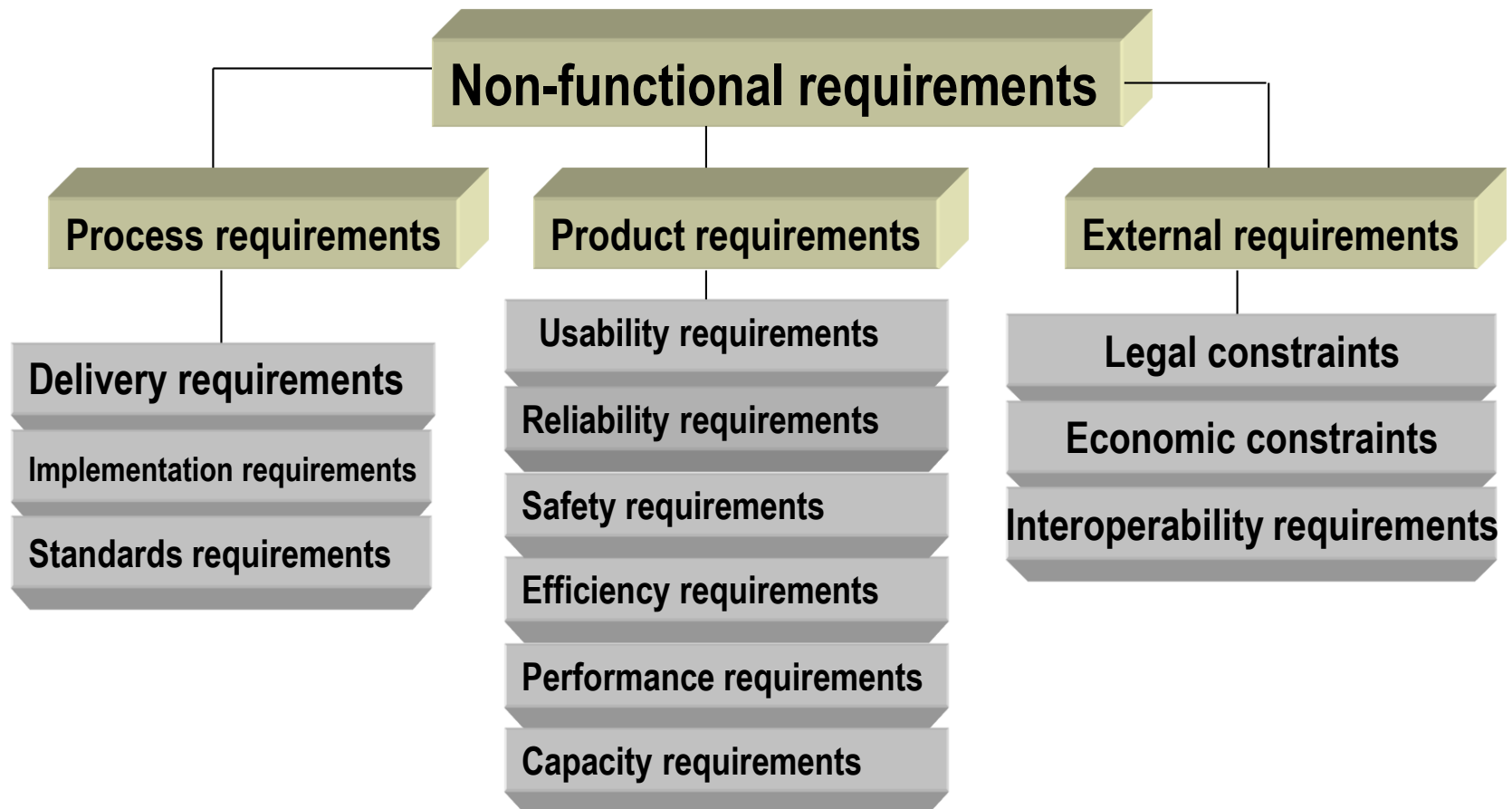
Source: See van Vliet 2000, pp111-3



IEEE Std-830 1998

Requisitos Específicos
Requisitos de performance
Requisitos de Interface
Requisitos Operacionais
Requisitos de Recursos
Requisitos de Verificação
Requisitos de Aceitação
Requisitos de Documentação
Requisitos de Segurança (security)
Requisitos de Portabilidade
Requisitos de Qualidade
Requisitos de Confiabilidade
Requisitos de de Manutenibilidade
Requisitos de <i>Safety</i>

Classificação de Sommerville



Requisitos de Produto

- São requisitos que especificam as características desejadas que um sistema deve fornecer
 - O serviço X do sistema deve ter disponibilidade de 99% (requisito de confiabilidade)
 - 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

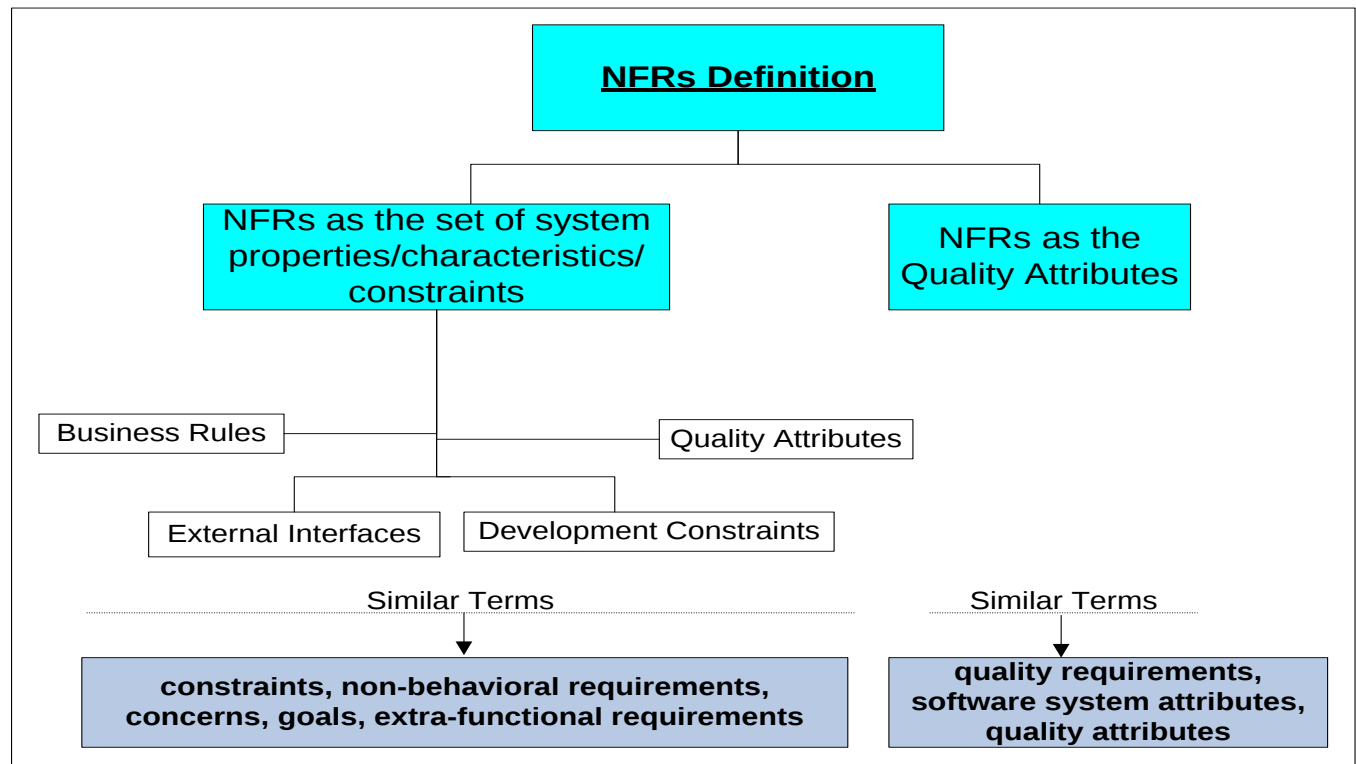
Requisitos Externos

- 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

1. Accessibility/Access Control
2. Accountability
3. Accuracy
4. Adaptability
5. Additivity
6. Adjustability
7. Affordability
8. Agility
9. Analyzability
10. Anonymity
11. Atomicity
12. Attractiveness
13. Auditability
14. Augmentability
15. Availability
16. Certainty
17. Changeability
18. Communicativeness
19. Compatibility
20. Completeness
21. Complexity/Interacting Complexity
22. Composability
23. Comprehensibility
24. Comprehensiveness
25. Conciseness
26. Confidentiality
27. Configurability
28. Conformance
29. Consistency

30. Controllability
31. Correctness
32. Customizability
33. Debuggability
34. Decomposability
35. Defensibility
36. Demonstrability
37. Dependability
38. Distributivity
39. Durability
40. Effectiveness
41. Efficiency/Device Efficiency
42. Enhanceability
43. Evolvability
44. Expandability
45. Expressiveness
46. Extendability
47. Extensibility
48. Fault/Failure Tolerance
49. Feasibility
50. Flexibility
51. Formality
52. Functionality
53. Generality
54. Immunity
55. Installability
56. Integratability
57. Integrity
58. Interoperability
59. Learnability

60. Legibility
61. Likeability
62. Localizability
63. Maintainability
64. Manageability
65. Maturity
66. Measurability
67. Mobility
68. Modifiability
69. Nomadicity
70. Observability
71. Operability
72. Performance/Efficiency/
Time or Space Bounds
73. Portability
74. Predictability
75. Privacy
76. Provability
77. Quality of Service
78. Readability
79. Reconfigurability
80. Recoverability
81. Reliability
82. Repeatability
83. Replaceability
84. Replicability
85. Reusability
86. Robustness
87. Safety

88. Scalability
89. Security/Control and Security
90. Self-Descriptiveness
91. Simplicity
92. Stability
93. Standardizability/
Standardization/Standard
94. Structuredness
95. Suitability
96. Supportability
97. Survivability
98. Susceptibility
99. Sustainability
100. Tailorability
101. Testability
102. Traceability
103. Trainability
104. Transferability
105. Trustability
106. Understandability
107. Uniformity
108. Usability
109. Variability
110. Verifiability
111. Versatility
112. Viability
113. Visibility
114. Wrappability

Tipos

- 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.

Tipos mais comuns

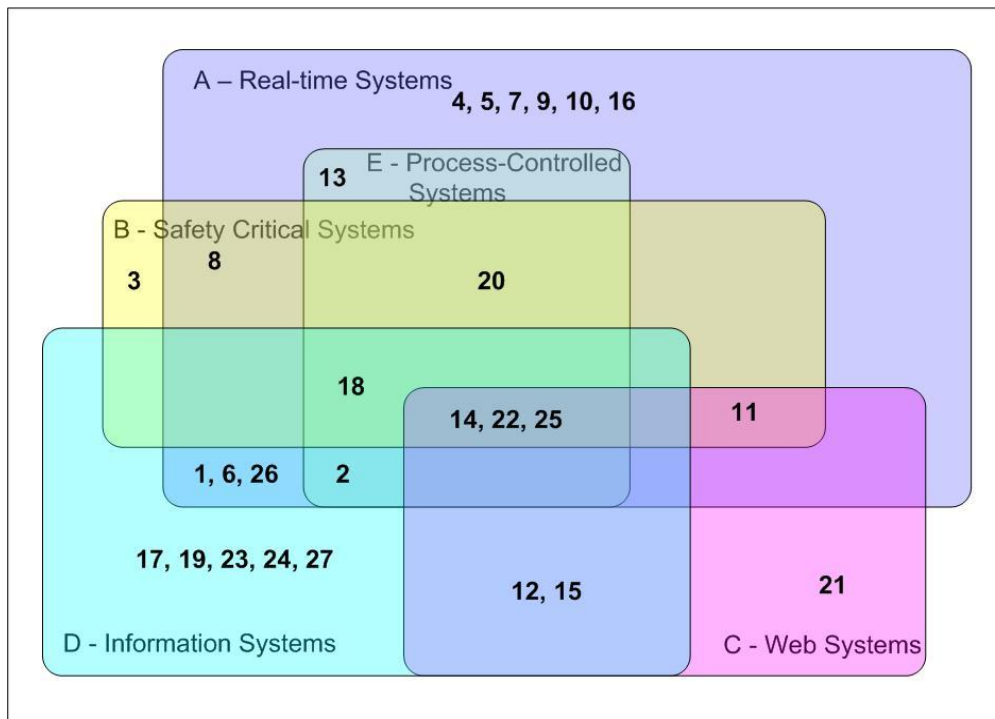
- Pq3: Que tipos de NFRs são mais comuns ou considerados na literatura?
- Os cinco mais frequentes tipos de NFRs:
 - performance (88.68%);
 - reliability (67.92%);
 - usability (62.26%);
 - security (60.38%);
 - maintainability (54.72%)

Tipos

- 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

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

The role of NFRs in Software Architecture [D. Ameller 14]

	First study	Second study
Type	Electronic survey	Interviews
Number of respondents	60	13
Number of RQs	5	6+7
Target population	Software industry	Software architects
Target information	Practical experience	Single project
Population origin	World-wide (>50% Spain)	Spain
Execution	2009	2010
Publication	2010	2012/13

Non-Functional Requirements in industrial practice

Half of respondents did not use NFRs to make architectural decisions

Respondents stated that:
“need tools for NFRs management”

Respondents stated that:
“want to have the last word on decision-making”

More empirical evidence for software architecture is needed

How do software architects deal with NFRs?

Companies did not have the role of architect clearly defined

NFRs were mostly elicited by the architects

Architects considered Non-technical NFRs as relevant as technical NFRs

Most of the architectural decisions had the influence of a NFR

NFRs Conclusions:

Architects take into account all kinds of requirements in architectural decisions

There is a wide space in the gap between researchers and practitioners

Replication and new empirical studies are required in this area

Derivando

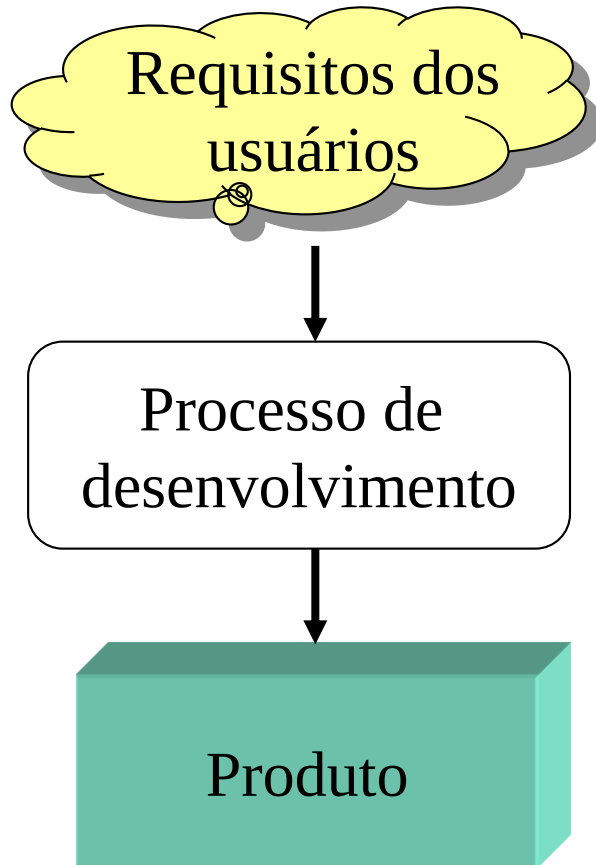
Requisitos não-funcionais

- ❑ Existem poucos métodos que tratam requisitos não-funcionais de forma efetiva.
- ❑ O motivo disso é porque eles são diversos e dependentes de cada domínio
- ❑ A maioria dos métodos de engenharia de requisitos é baseada na análise funcional ou orientada a objetos

Problemas para expressar requisitos não-funcionais

- ❑ Certas restrições, por exemplo tempo de resposta a falhas, são relacionadas a soluções de projeto desconhecidas no estágio de requisitos
- ❑ Requisitos não-funcionais normalmente estão relacionados a um ou mais requisito funcional
- ❑ Não existem regras que determinem quando um NFR está completamente atendido

Tratamento de requisitos não-funcionais



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 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

Métricas de Qualidade

□ Passos gerais para usar métricas:

1. Determine o conjunto de atributos desejáveis (RNFs)
2. Determine a importância relativa de cada atributo
3. Avalie a conformidade do sistema em relação aos atributos
4. Calcule a pontuação obtida em cada atributo e a qualidade geral do sistema

Métricas de Qualidade

Atributo	Métrica
velocidade	Transações/seg, tempo de resposta
Facilidade de uso	Linguagem usada, interface gráfica, help online
tamanho	Kbytes, LOCs, Pontos de função, medidas de complexidade

Avaliação da qualidade geral do sistema

Atributo	Peso relativo	Escore de conformidade	Escore final
Velocidade	.3	6	1.8
Facilidade de uso	.6	5	3.0
tamanho	.1	7	0.7
Qualidade geral			5.5/10

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

Comparação entre as abordagens orientadas a processo e a produto

- ❑ Não existe uma abordagem melhor que a outra, elas são complementares e devem ser usadas para obter sistemas que de fato atendam aos requisitos não-funcionais dos stakeholders
- ❑ Durante o estágio inicial de análise de requisitos é recomendável usar abordagens de processo
- ❑ Abordagens orientadas a produto são indicadas quando os requisitos estão bem definidos e podem ser especificados em termos de funcionalidades e fatores qualitativos mensuráveis

O Framework NFR

- Proposto por **Chung**, University of Toronto
- Representação sistemática e global de NFRs
- Abordagem orientada a processo
- Abordagem qualitativa
- Representa NFR explicitamente como softgoals

Principais características

- **Softgoals** - são unidades básicas para representar requisitos não-funcionais
- **Interdependências** – estabelecem relacionamentos entre softgoals
- **Métodos** – oferecem técnicas de operacionalização
- **Correlações** – oferecem catálogos para inferir possíveis interações

Softgoals

- É uma meta que não possui definição clara
- Suporta raciocínio qualitativo e graus de satisfação
- Interage em sinergia ou conflito
- Decomposto através de relacionamentos **AND** ou **OR**
 - **AND** – o softgoal é satisfeito se todos os seus sub-softgoals são
 - **OR** - o softgoal é satisfeito se qualquer um dos seus sub-softgoals são
- São atendidos, ao invés de satisfeitos

Framework NFR

□ Notação

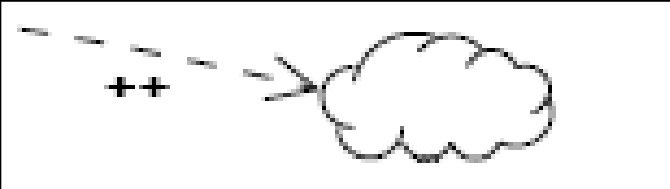
	AND-Decomposition
	OR-Decomposition
	Operationalization

Softgoals

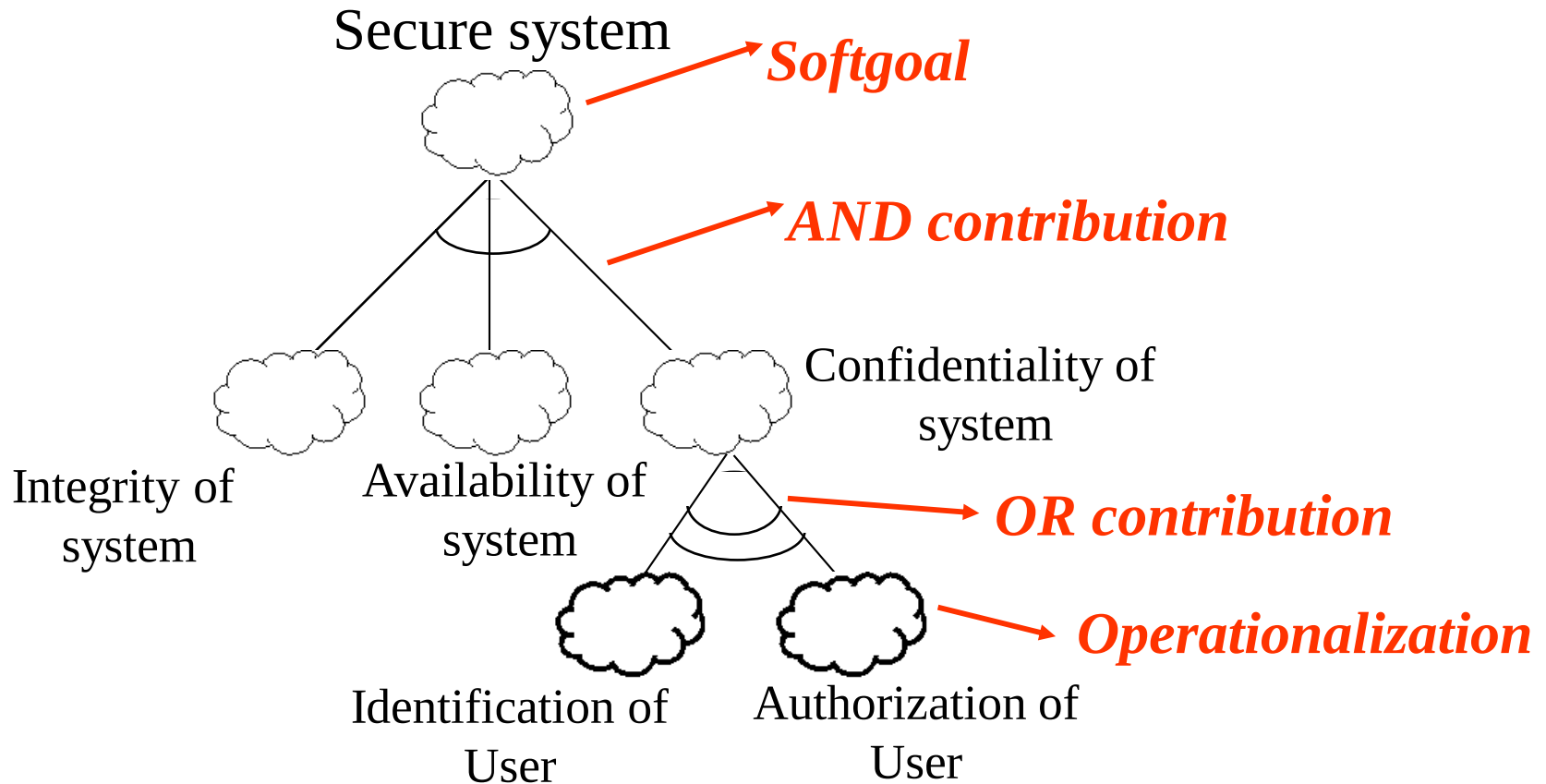
- Geralmente não admitem um consenso na sua definição
- São inter-relacionados e frequentemente conflitantes
- Define quão bem o sistema cumpre as suas funções
- Podem ser vistos com “fuzzy goals”, pois não possuem um critério claro de atendimento

Contribuições de softgoal

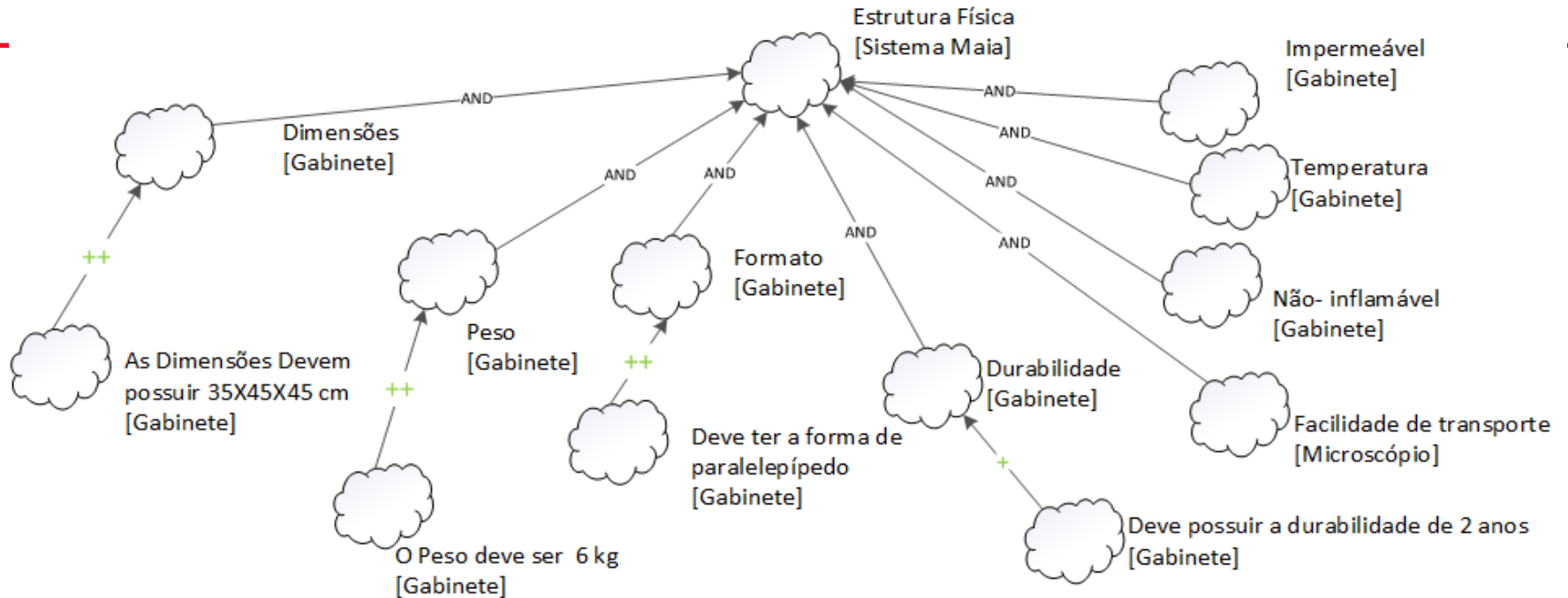
- Modelam influência positiva suficiente/parcial (respectivamente ++ e +) ou negativa suficiente/parcial (respectivamente - - e -) para atender softgoals

	partial/positive
	sufficient/positive
	partial/negative
	sufficient/negative

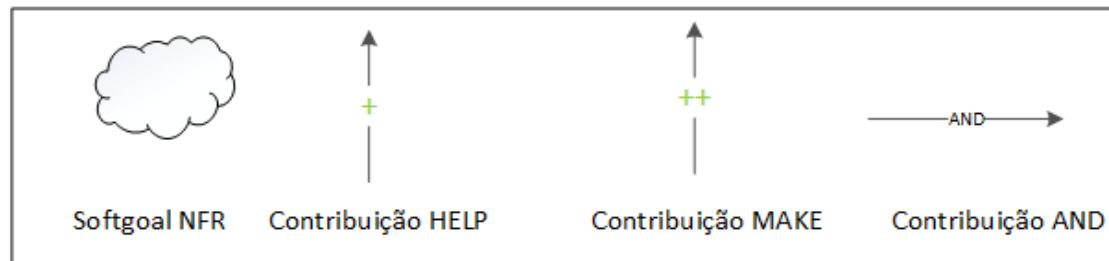
Softgoal Interdependency Graph (SIG)



SIG - Softgoal Interdependency Graph



Legenda

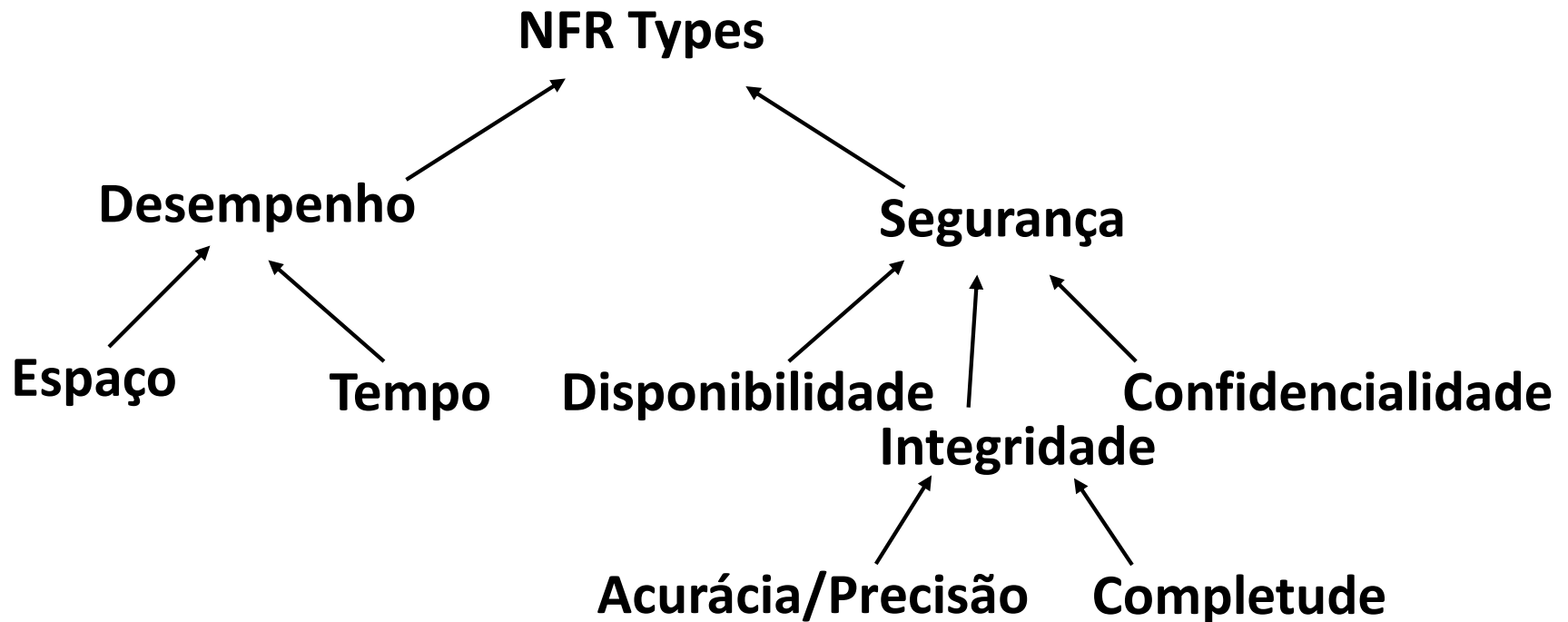


Fonte:Próprio Autor

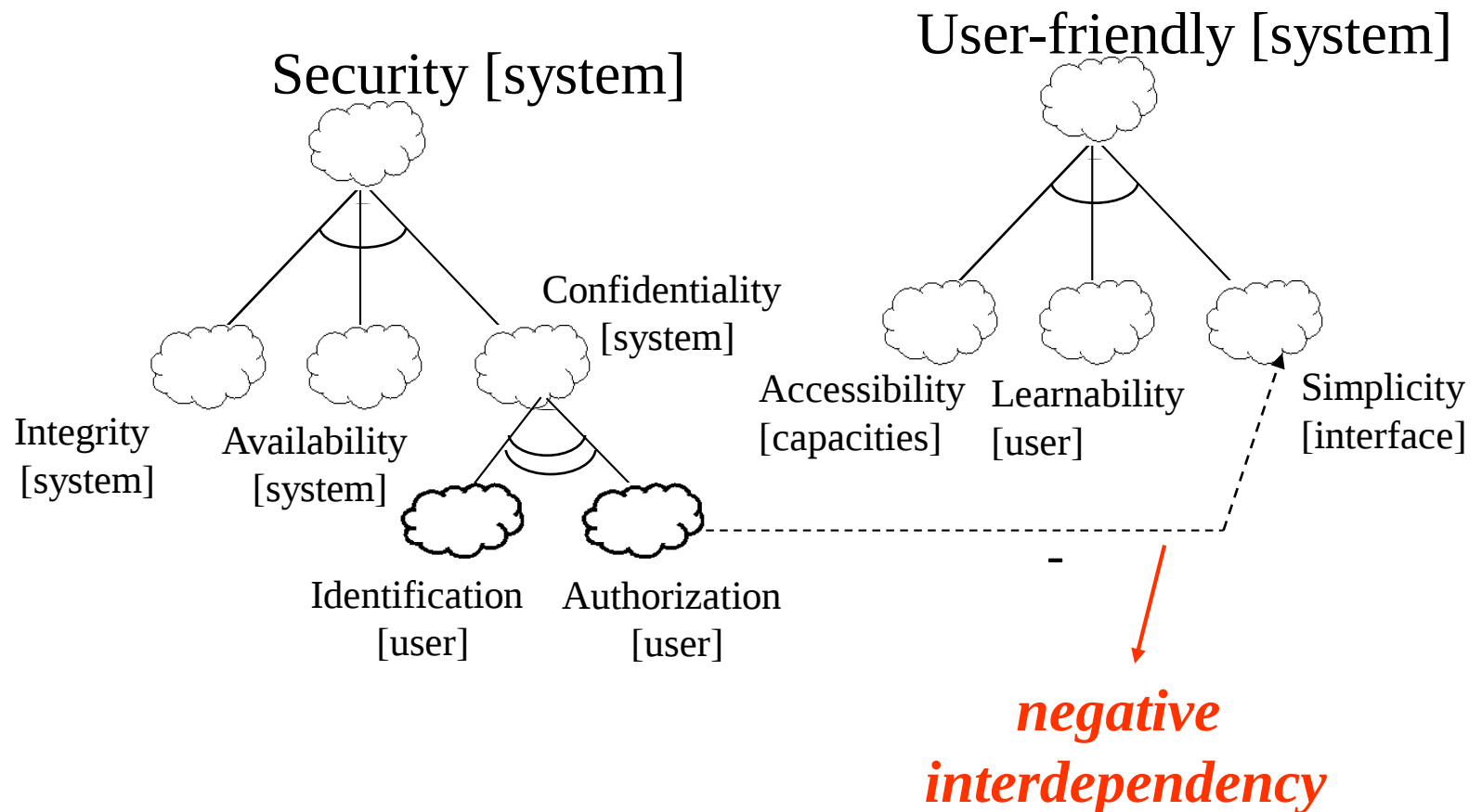
Catálogos

- ❑ Apresentam conhecimento sobre NFRs
- ❑ Fontes de conhecimento são:
 - especialistas no domínio, desenvolvedores, livros-texto, guias de desenvolvimento, etc.
- ❑ Tipos de catálogos:
 - Tipos de NFR (organiza NFRs em hierarquias organizadas)
 - método (refina NFRs considerando operacionalizações)
 - correlação (mostra interdependências implícitas)

Catálogo de alguns tipos de NFRs



Interdependências implícitas no SIG

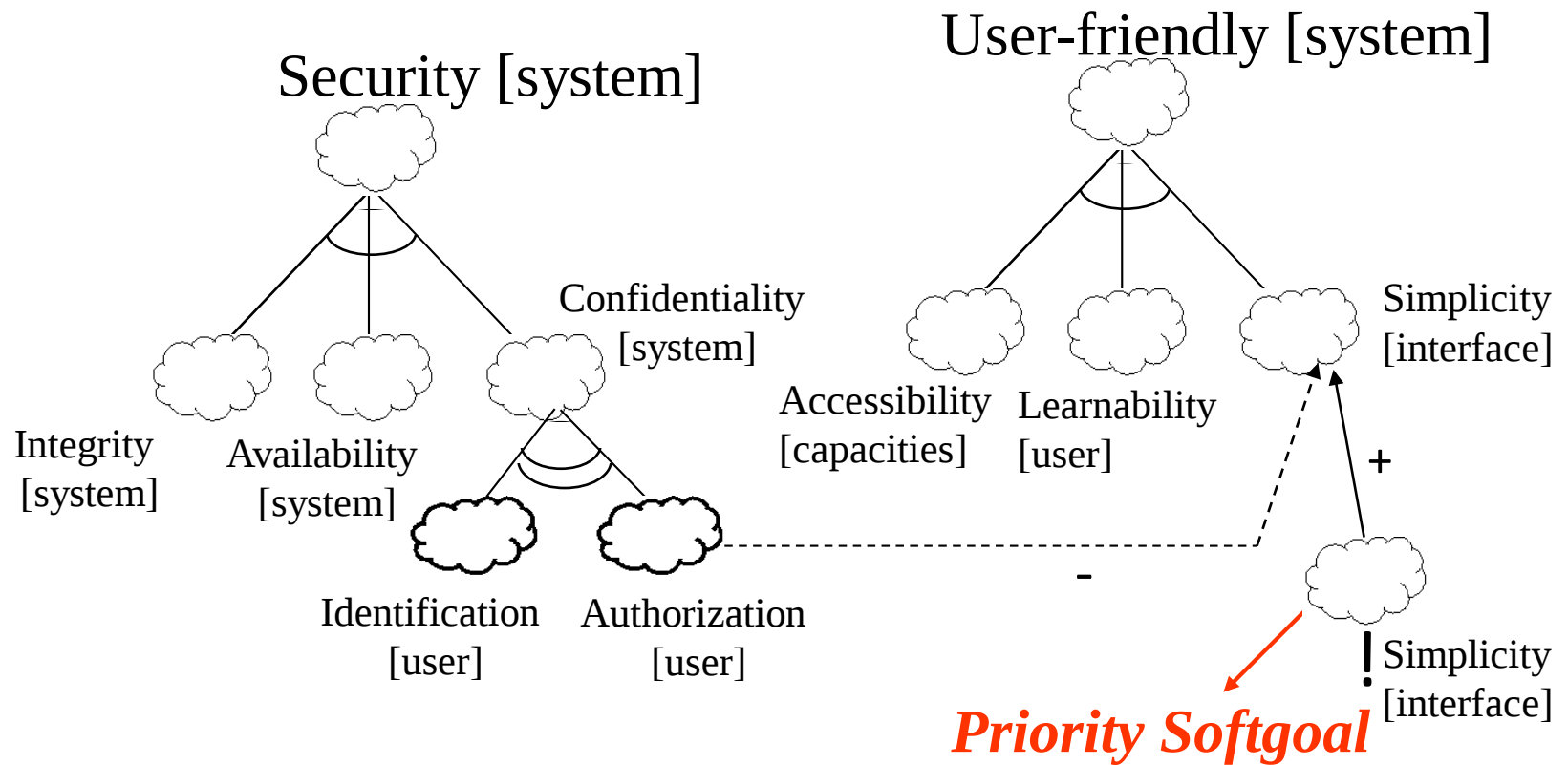


Prioridades

- Prioridade de softgoals podem ser identificados como:
 - Crítica – vital para o sucesso do sistema
 - Dominante – lida com uma parte significativa da carga de trabalho da organização

- Ajuda a resolver conflitos (tradeoffs) entre softgoals

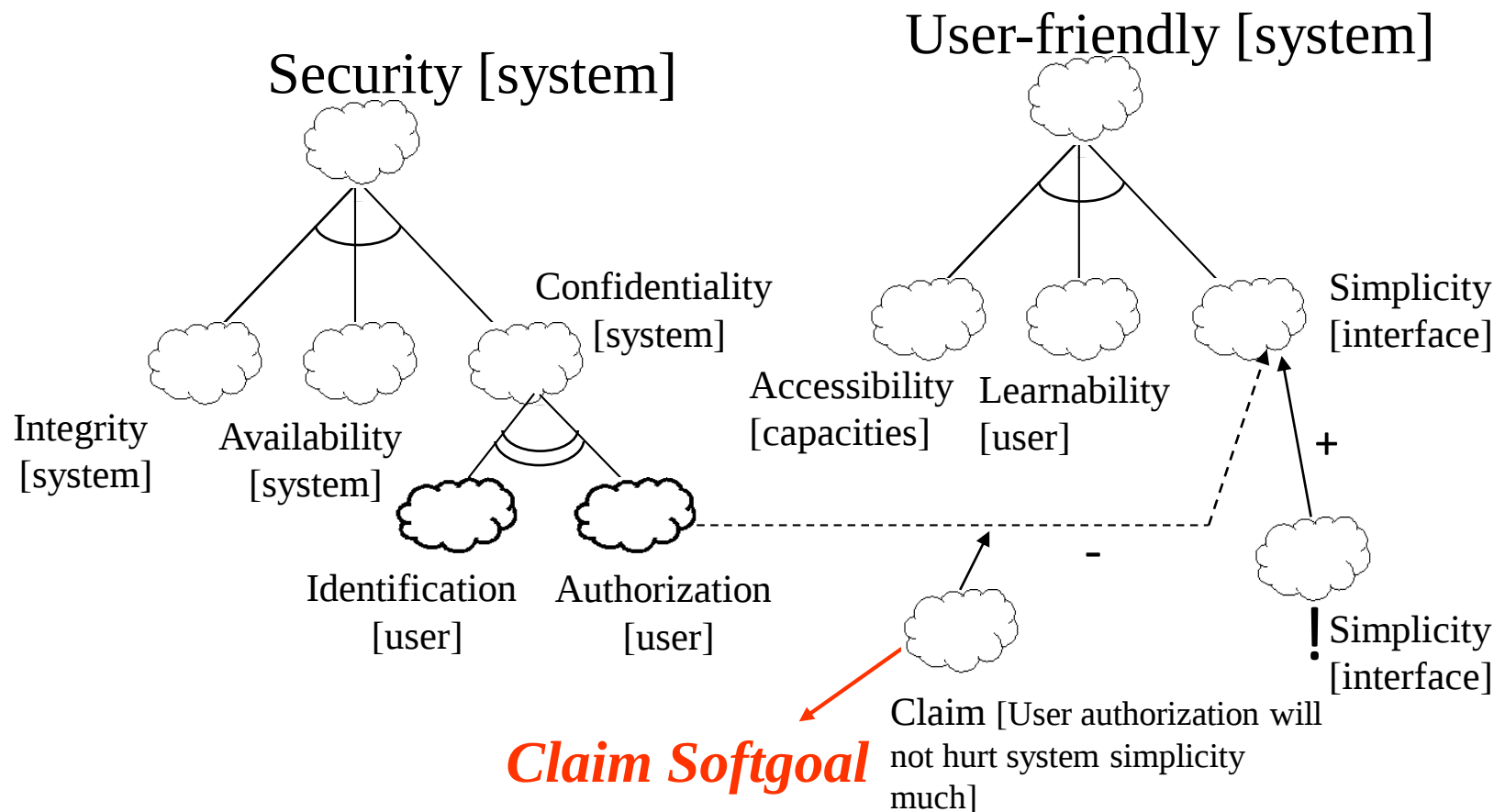
Identificando Prioridades



Registrando Raciocínio de Projeto

- Decisões de projeto deveriam ser suportadas por argumentos bem justificados
- Razões podem ser estabelecidas através de refinamentos, pela seleção de alternativas, etc.
- Um softgoal “Claim” pode racionalizar os tradeoffs

Registrando Raciocínio de Projeto



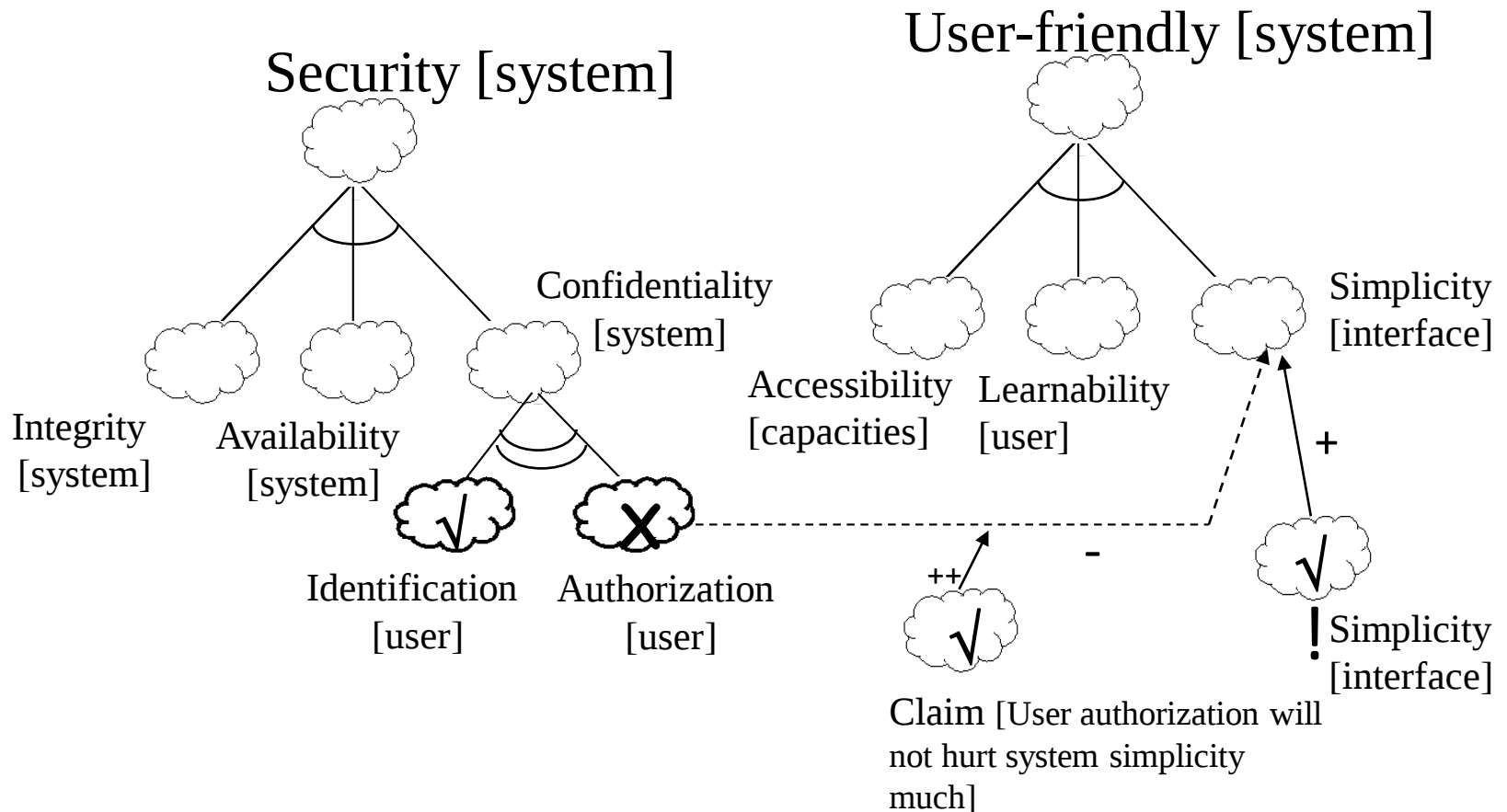
Selecionando entre alternativas

- O processo de refinamento continua até que as possíveis soluções sejam suficientemente detalhadas
- Avalia o impacto das decisões
- Consideram operacionalizações e decidem se uma alternativa escolhida atende suficientemente um softgoal

Avaliando o impacto de decisões

- Processo Bottom-up
- Avaliação de softgoals são representadas por rótulos (tais como $\surd$ e X)
- Contribuição positiva
 - Filhos “atendidos” resultam em pais “atendidos”
 - Filhos “negados” resultam em pais “negados”
- Contribuição negativa
 - Filhos “atendidos” resultam em pais “negados”
 - Filhos “negados” resultam em pais “atendidos”

Seleccionando entre alternativas



Framework NFR

- Estabelece interdependencias entre NFRs
- Lida com *tradeoffs* (conflitos) e prioridades
- Seleciona operacionalizações
- Suporta decisões (design rationale)
- Avalia o impacto das decisões

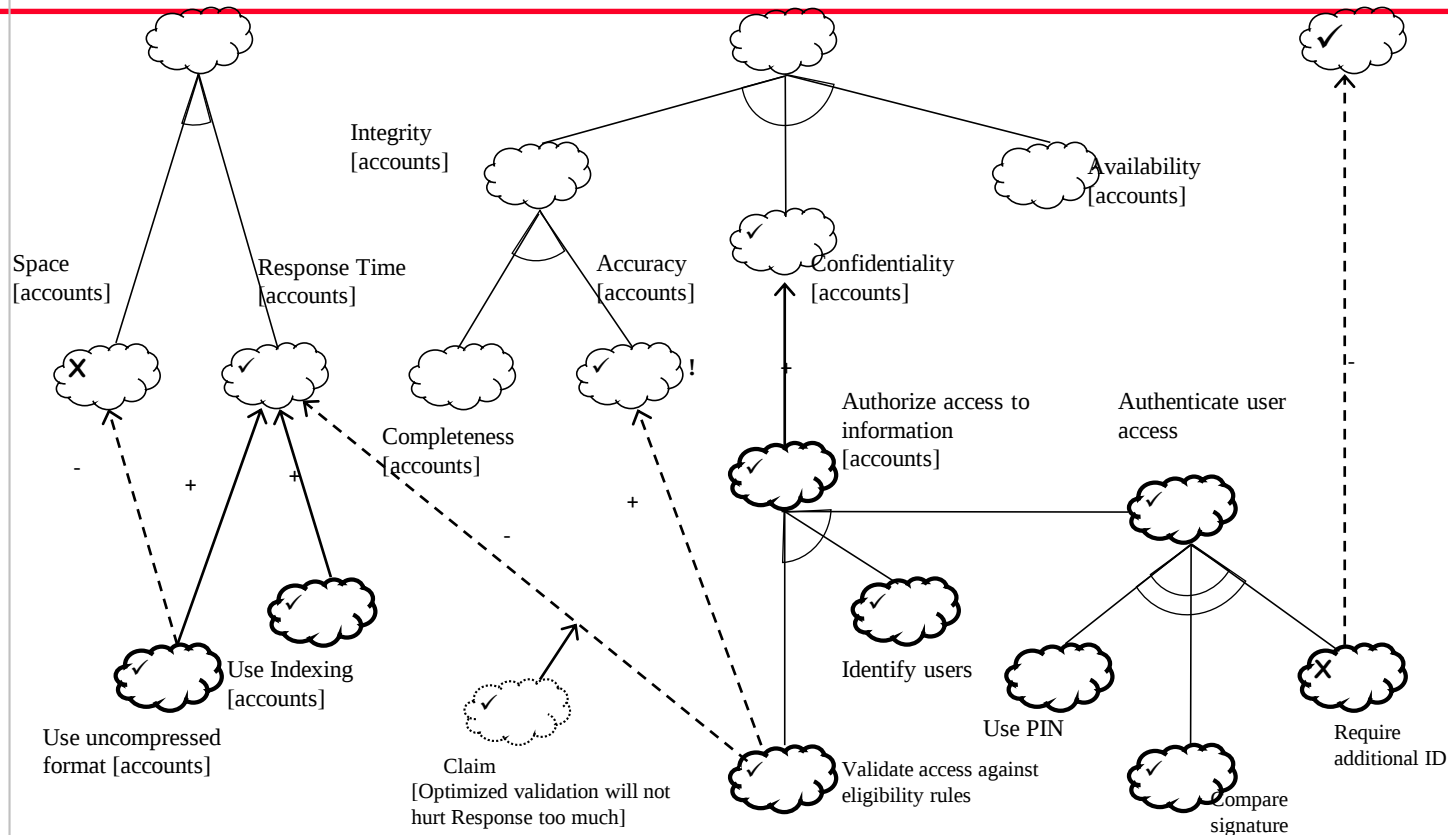
Banking System

[CHUNG et al. 2000]

Good Performance
[accounts]

Secure
[accounts]

User-Friendly Access
[accounts]



Softgoals



NFR Softgoal



Operationalizing Method



Claim



Critical

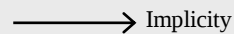


Accepted



Rejected

Interdependency



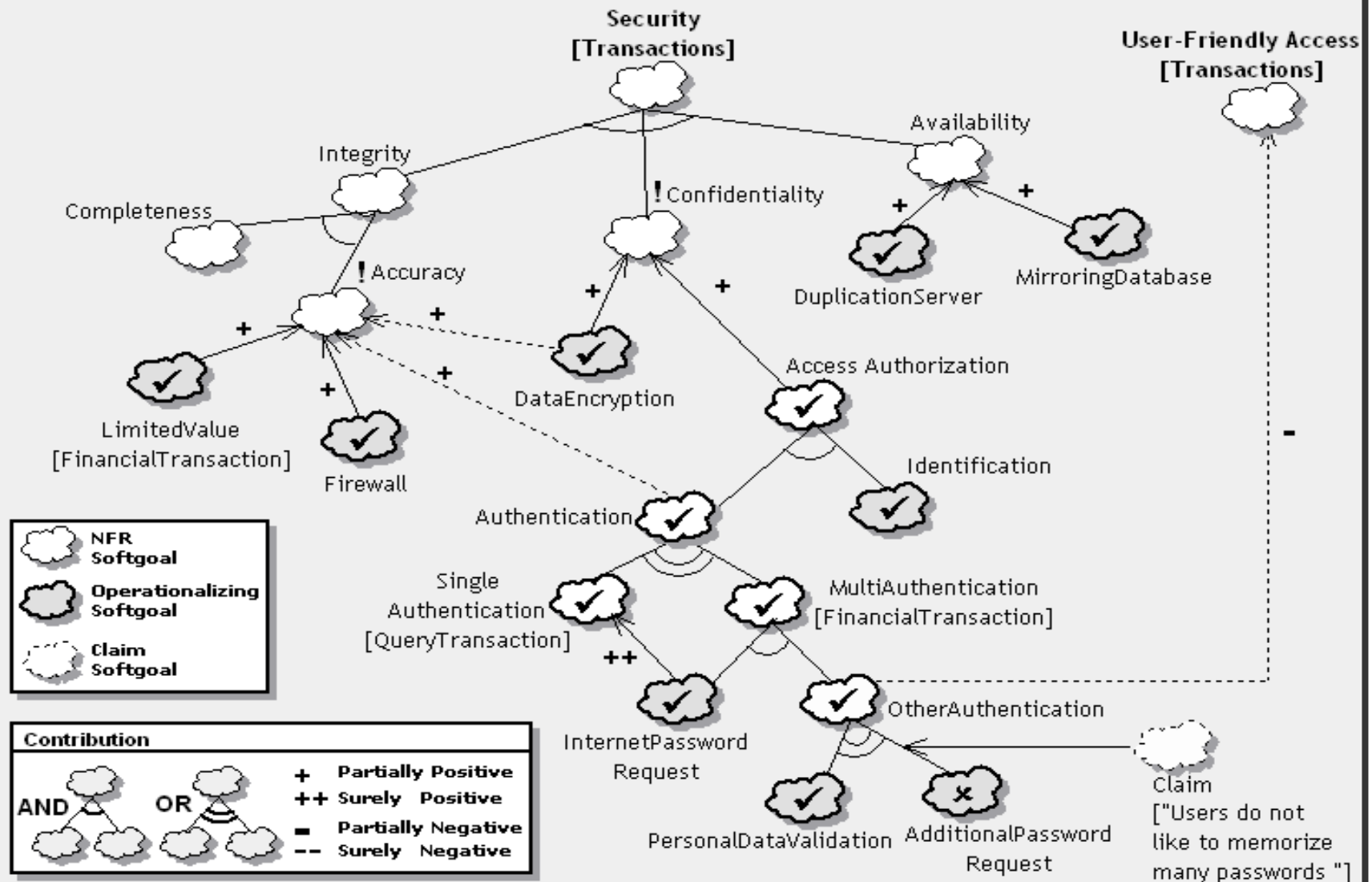
Implicit



Explicit

++ Strongly positive satisficing
+ Positive satisficing
- Negative satisficing
-- Strongly Negative satisficing

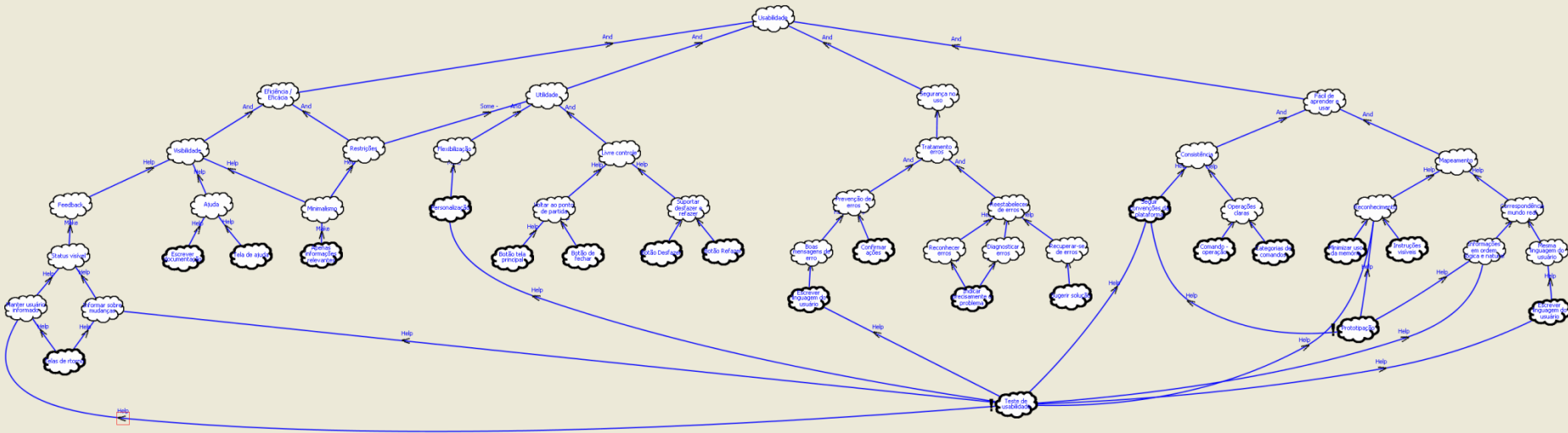
Internet Banking System



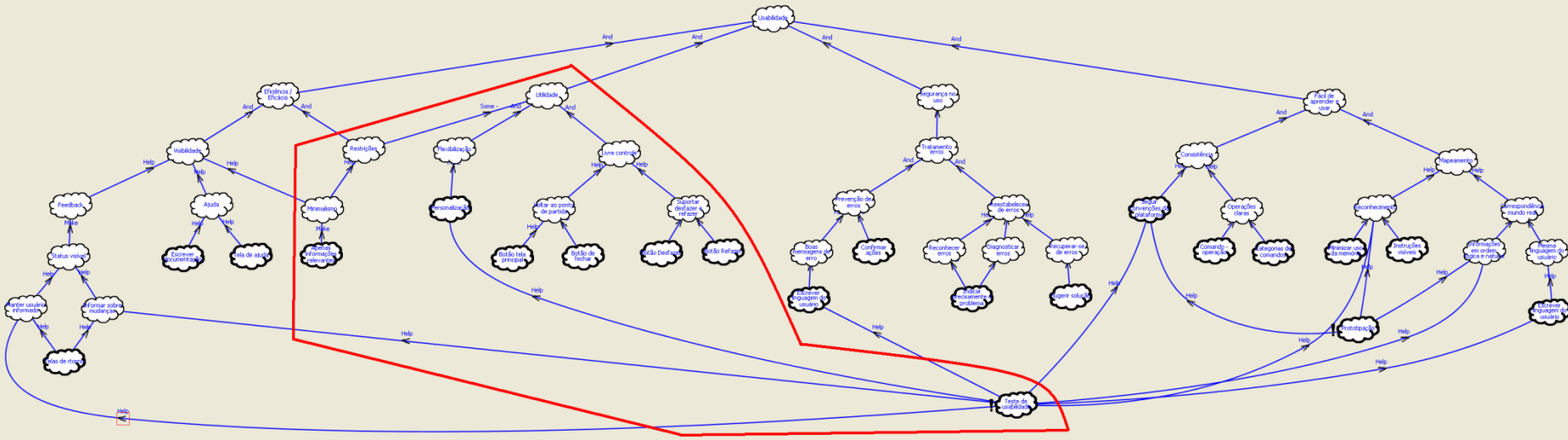
Catálogo de Usabilidade [L. Xavier , 2009]

[L. Xavier , 2009]

- Colocar catálogo Completo

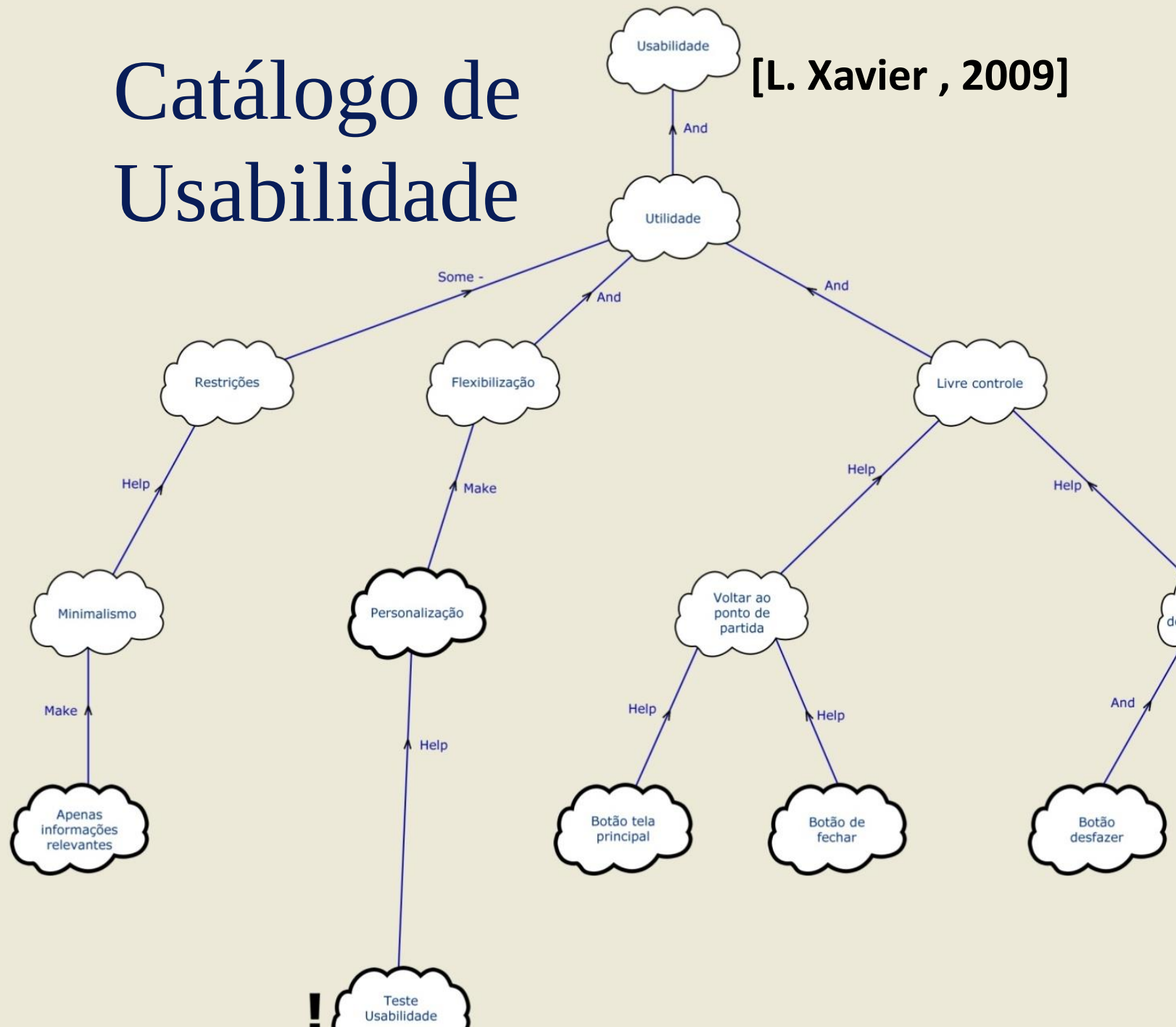


- Colocar catálogo Completo



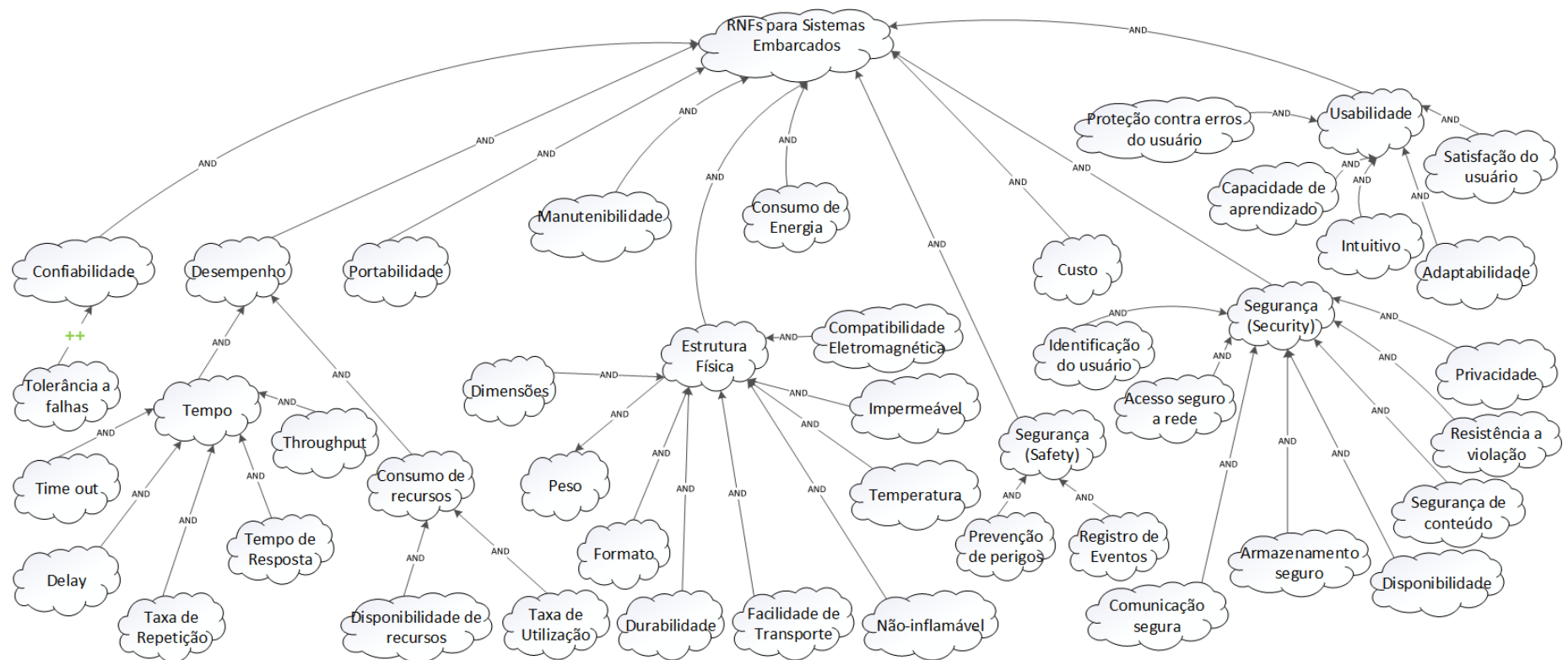
Catálogo de Usabilidade

[L. Xavier , 2009]

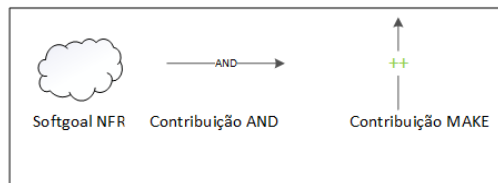


O Catálogo NFR4ES (Sem Correlações)

[R. Silva , 2019]

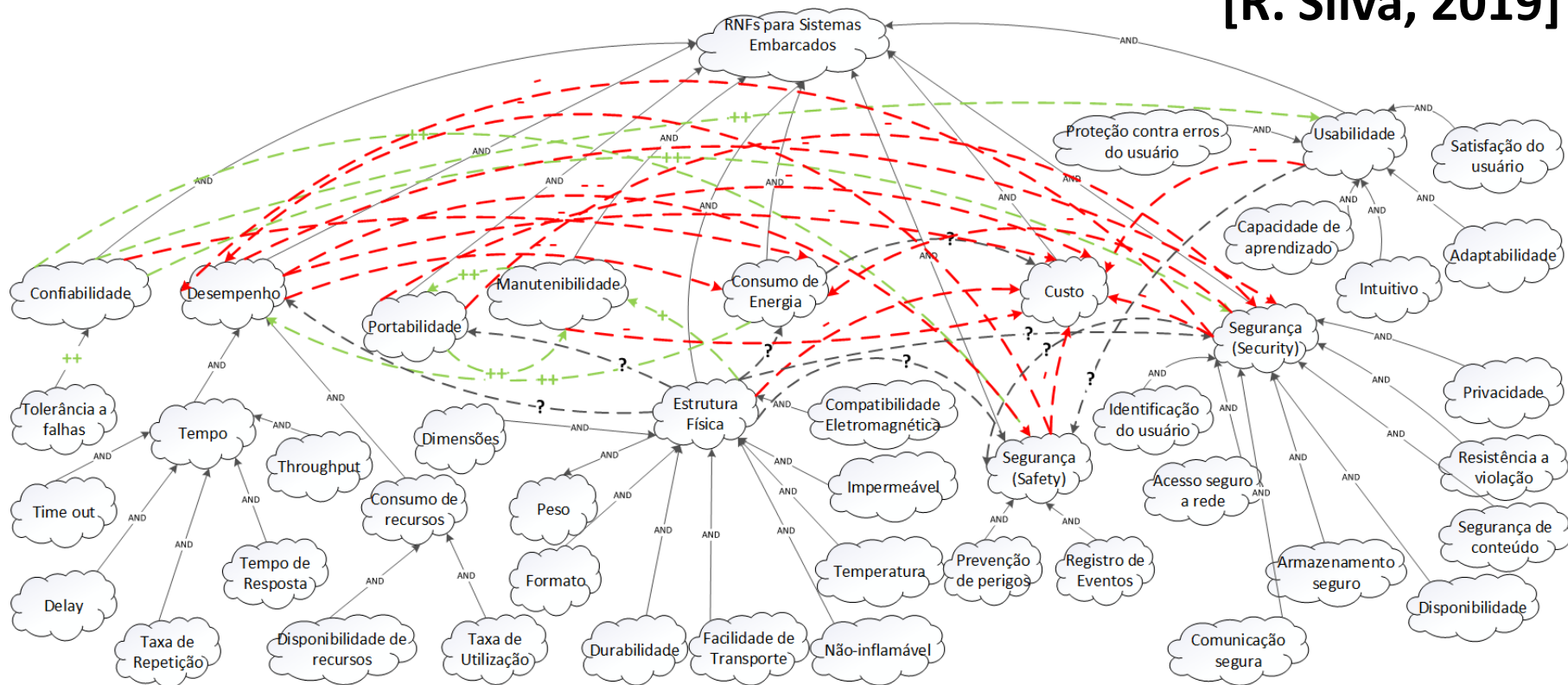


Legenda

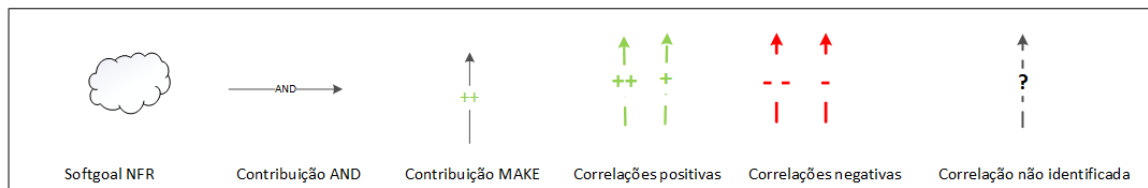


O Catálogo NFR4ES (Com Correlações)

[R. Silva, 2019]



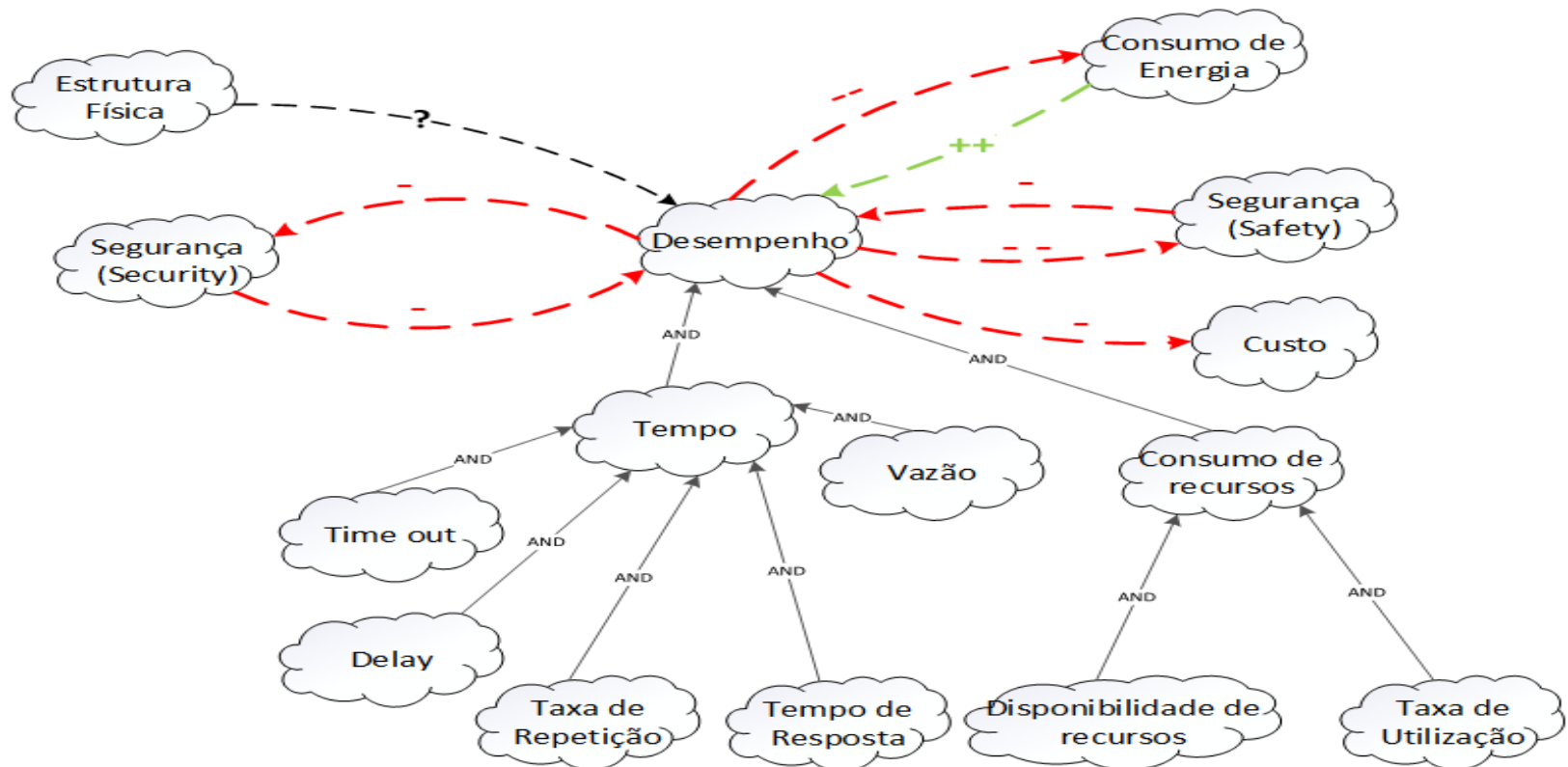
Legenda



NFR4ES: Desempenho

[R. Silva, 2019]

- Definição: são requisitos que definem uma demanda operacional em relação a tempo de computação e consumo de recursos (BROY; SLOTOCH, 2001).

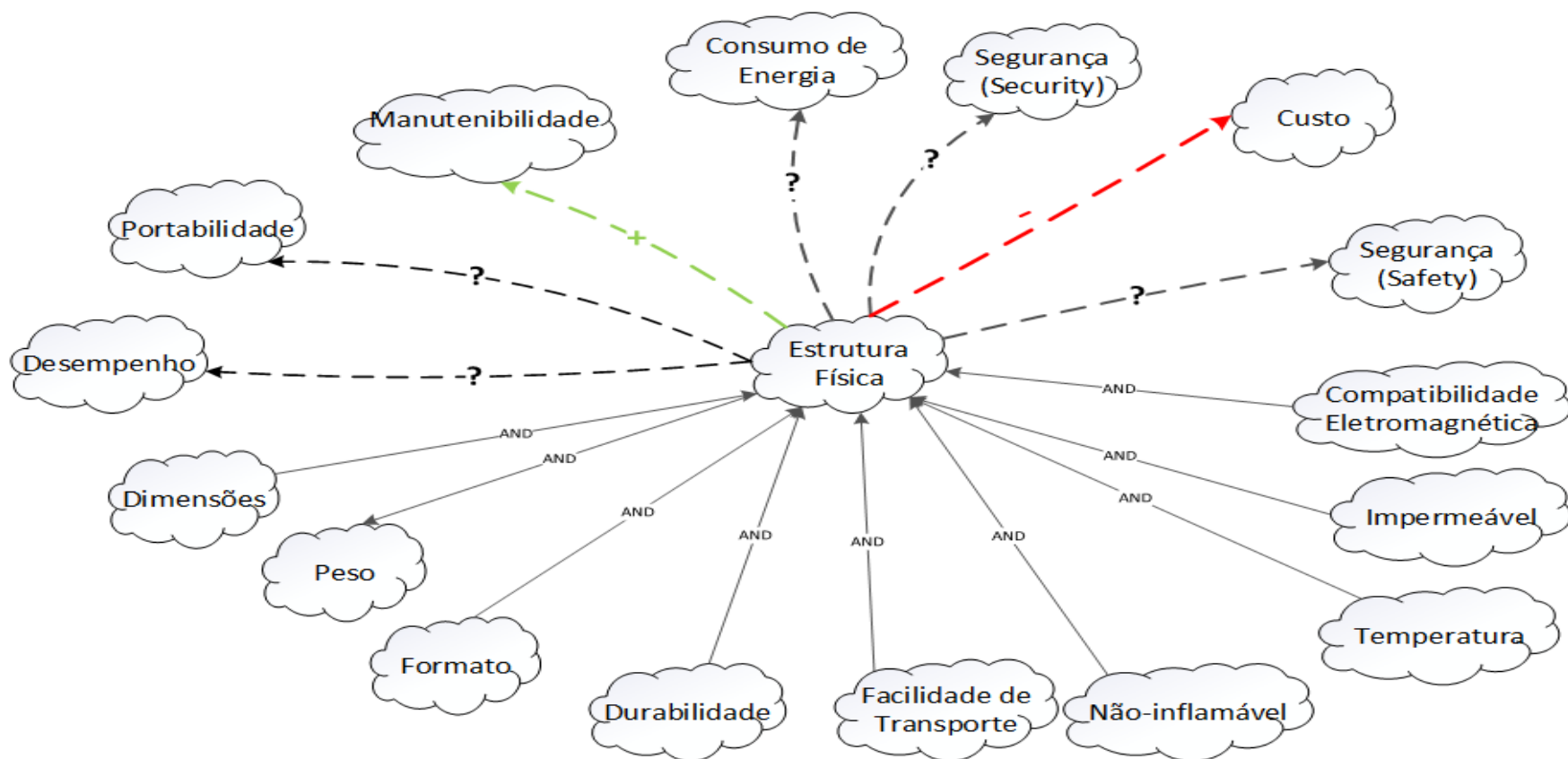


Fonte:Próprio Autor

NFR4ES: Estrutura Física

[R. Silva, 2019]

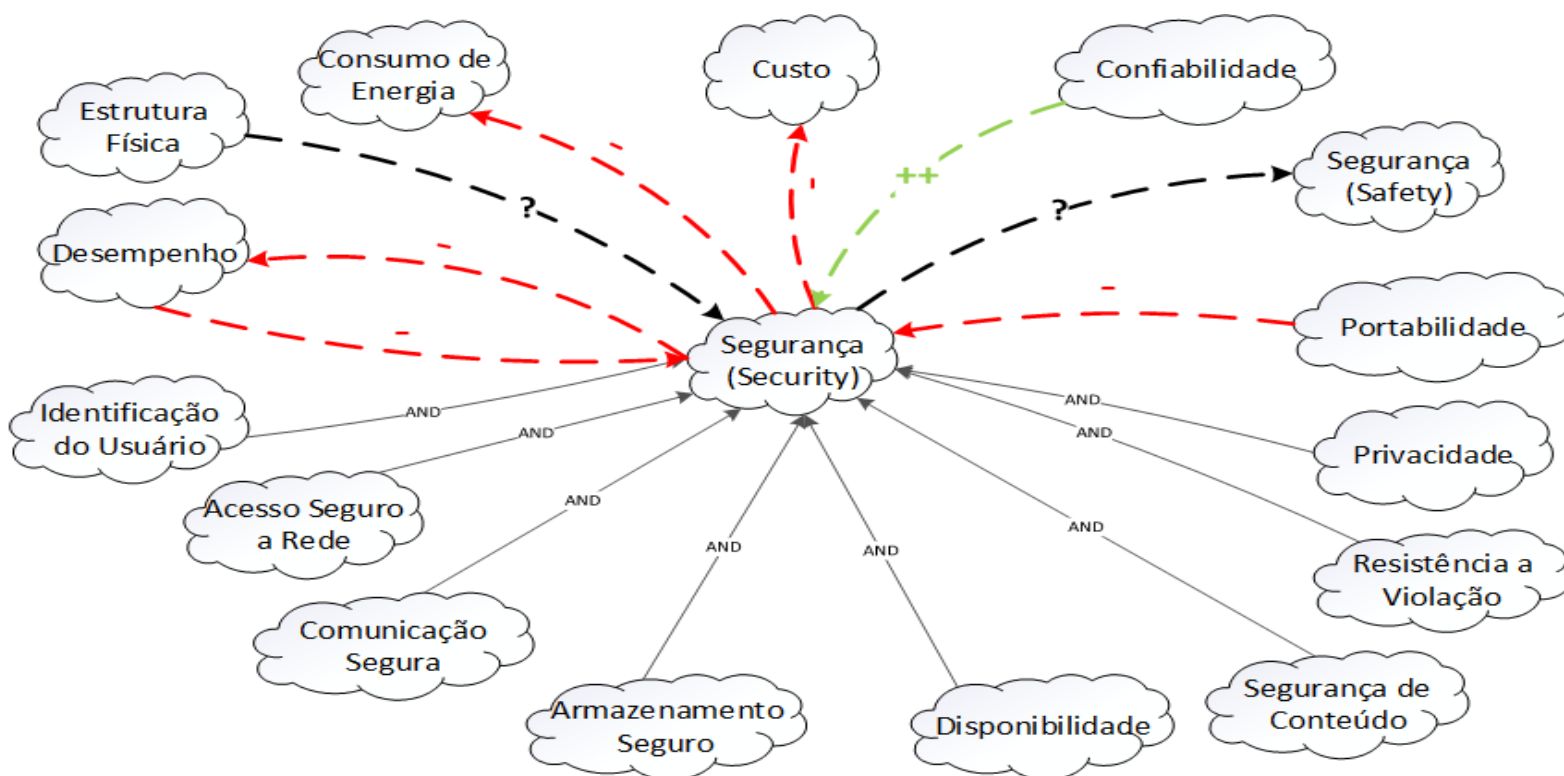
- Definição: são requisitos que expressam as características físicas sobre os componentes do sistema (WOLF, 2012).



Fonte: Próprio Autor

NFR4ES: Segurança (Security) [R. Silva, 2019]

- Definição: são requisitos que se concentram na capacidade do sistema em se proteger contra invasões acidentais ou deliberadas (LI; MA; YAO, 2015).



Fonte: Próprio Autor

Pontos Principais

- O uso do NFR Framework pode ser visto como uma análise de requisitos que trata de forma detalhada os requisitos não-funcionais do sistema
- Inicialmente, os requisitos são decompostos em em requisitos mais específicos, onde são tratadas ambigüidades e prioridades entre eles, ao longo desse processo são identificadas interdependências entre os requisitos não-funcionais

Pontos Principais

- Em seguida, são consideradas as diversas alternativas que podem ser usadas no projeto e implementação do sistema, onde algumas delas são escolhidas. Nesse momento deve-se mostrar as razões para tomar tais decisões
- Finalmente, examina-se quais requisitos não-funcionais puderam ser satisfeitos e relaciona-se as decisões obtidas com o uso do NFR Framework com os requisitos funcionais que devem ser implementados