



REQUISITOS FUNCIONAIS Vs. REQUISITOS NÃO FUNCIONAIS

1

REQUISITOS

- Requisitos são capacidades e condições às quais o sistema (e o projeto) devem responder
- *It is certainly a myth that the requirements for large software projects are ever perfectly understood or perfectly specified* (Abran et al., 2001)
- O nível de detalhe da especificação de requisitos geralmente aumenta ao longo do tempo de um projeto, pelo menos para alguns dos requisitos

REQUISITOS NÃO FUNCIONAIS NOS CASOS DE USO?


- Podem ser capturados nos UC se relacionados apenas com um UC particular
- Na secção “requisitos especiais” dum UC podem ser identificados:
 - requisitos não funcionais
 - atributos de qualidade
 - restrições específicas
- E.g. “A informação monetária (e.g. custo estimado de uma tarefa) é indicada em POTs (moeda virtual interna à plataforma)” pode ser capturado na secção “requisitos especiais” do caso de uso

TECNOLOGIA E LISTA DE VARIAÇÕES DOS DADOS (UC – FORMATO COMPLETO)

- Variações técnicas na forma como algo é feito, e muitas vezes essas dizem respeito à entrada/saída de dados
 - E.g. tecnologia/standards/recomendações relacionadas com a utilização dum leitor de cartões
- Esquemas de codificação que serão utilizados para alguns dados
 - E.g. Standard/regra de codificação do identificador de produtos em UPC (Universal Product Code) é muito utilizada em várias lojas
- Alguma alteração agora? Algo deve ser adicionado a algum caso de uso?

ONDE ESPECIFICAR OS REQUISITOS - REGRA

- Requisitos específicos do UC são documentados no UC
- TODOS OS OUTROS são documentados na Especificação Suplementar (cf. a seguir)



Onde se especificam?

“A velocidade de rotação das rodas do veículo deve ser medida 15 vezes por segundo para que o ABS opere com segurança.”

“A interação entre os utilizadores e o sistema deve ser simples, intuitiva e completamente adaptada à ação em causa.”

“A aplicação deve ser implementada na linguagem de programação java.”



ESPECIFICAÇÃO SUPLEMENTAR

6

ESPECIFICAÇÃO SUPLEMENTAR

- “Supplementary Specification”
- Capta
 - Requisitos não funcionais (e.g. documentação, empacotamento, suporte, licenciamento, etc.)
 - Requisitos funcionais não capturados nos UC
- Adota o modelo FURPS+

The left side of the slide features a series of vertical stripes in shades of brown, tan, and grey. Overlaid on these stripes are several orange circles of varying sizes. One large circle is positioned near the top left, and several smaller circles are scattered below it, some overlapping the stripes.

8

Modelo FURPS+

MODELO FURPS+

Português	Inglês
Funcionalidade*	Functionality*
Usabilidade	Usability
Confiabilidade	Reliability
Desempenho	Performance
Suporte	Supportability
+: {implementação interface, operações, empacotamento, legal}	+: {implementation, interface, operations, packaging, legal}

Requisitos de
qualidade

Requisitos
não funcionais

FURPS+ - FUNCIONALIDADE

- Funcionalidades tipicamente não capturadas nos casos de uso:

Função	Descrição/exemplo
Auditoria	Registo de dados adicionais sobre a execução do sistema
Licenciamento	Existência de serviços relacionados com a aquisição, instalação e monitorização da licença do software
Locais/línguas	Possibilidade de múltiplas línguas ou outros aspectos relacionados com a utilização do software em diferentes pontos geográficos
Correio (eletrónico)	Existência de serviços relacionados com envio/recepção de correio

FURPS+ - FUNCIONALIDADE (CONT.)

Função	Descrição/exemplo
Ajuda	Existência de suporte informativo para os utilizadores do sistema
Impressão	Possibilidade de impressão de dados relacionados com a atividade do sistema
Geração de relatórios	Suporte para a geração de relatórios
Segurança	Acesso controlado a determinadas funcionalidades do sistema
Gestão do sistema	Existência de serviços que facilitam a gestão de aplicações num ambiente distribuído
Workflow	Suporte para a gestão do estado de itens de trabalho (e.g., o que está aprovado, pendente)

Adaptado de (Eeles, 2005).

FURPS+ - FUNCIONALIDADE (EXEMPLO)

○ Exemplos:

- *(Segurança)*

As interações dos utilizadores supramencionados devem ser precedidas de um processo de autenticação.

- *(Segurança)*

A utilização da plataforma por outras pessoas é restrita ao registo de organizações.

FURPS+ - USABILIDADE

- Avalia a interface com o utilizador
- Possui diversas subcategorias, entre elas:
 - Prevenção de erros inseridos pelo utilizador
 - Adequação da interface aos diversos tipos de utilizadores
 - Estética e design
 - Consistência da interface

FURPS+ - FIABILIDADE/CONFIABILIDADE

- Refere-se a:
 - Integridade,
 - Conformidade,
 - Interoperabilidade do software
- Os requisitos que podem ser considerados são:
 - frequência e gravidade de falhas do sistema
 - possibilidade de recuperação de falhas
 - exatidão de cálculos

FURPS+ - DESEMPENHO

- Avalia os requisitos de desempenho do software, nomeadamente:
 - tempo de resposta
 - tempo de inicialização (*setup*) do sistema
 - tempo de término (*shutdown*) do sistema
 - consumo de memória
 - utilização da CPU
 - capacidade de carga

FURPS+ - SUPORTABILIDADE

- Os requisitos de suportabilidade agrupam várias características, como:
 - testabilidade
 - adaptabilidade
 - manutenibilidade
 - compatibilidade
 - configurabilidade
 - instalabilidade
 - escalabilidade

FURPS+ - SUPORTABILIDADE (EXEMPLO)

- Exemplo:

- *(configurabilidade)*

“A designação comercial da plataforma e outros dados que venham a ser relevantes devem ser especificados por configuração aquando da sua implantação.”

FURPS+ - OUTROS (+)

- Restrições de design - especifica ou restringe o processo de design do sistema.
- Exemplos:
 - *Adotar boas práticas de identificação de requisitos e de análise e design de software OO.*
 - *Reutilizar o componente de gestão de utilizadores existente na T4J baseado em identificador de utilizador (i.e. email) e palavra-passe (cf. documentação).*

FURPS+ - OUTROS (+)

- Restrições de implementação - especifica ou restringe o código ou a construção de um sistema
- Exemplos:
 - *Implementar o núcleo principal do software em Java.*
 - *Adotar normas de codificação reconhecidas.*



EXEMPLO/EXERCÍCIO

20

ESPECIFICAÇÃO SUPLEMENTAR: TRABALHO PRÁTICO (ITERAÇÃO 2)

Funcionalidades

?

Usabilidade

?

Fiabilidade/Confiabilidade

?

Desempenho

?

ESPECIFICAÇÃO SUPLEMENTAR: TRABALHO PRÁTICO (ITERAÇÃO 2)

Suportabilidade

?

Outras (+)

Restrições de design

?

Restrições de implementação

?

BIBLIOGRAFIA

- Craig Larman. *Applying UML and Patterns* (3rd ed.); 2002.
- Abran, A., Bourque, P., Dupuis, R., & Moore, J. W. (2001). Guide to the software engineering body of knowledge-SWEBOK. IEEE Press.
- Eeles, P. (2005). Capturing architectural requirements. Disponível em <http://www.ibm.com/developerworks/rational/library/4706.html>.