

Taxonomia de Requisitos

ou

Criando uma padronização nas definições para os componentes de uma especificação pelo conhecimento adquirido em Engenharia de Requisitos via real aplicabilidade por Paulo Nannini

Baseado na apresentação de Martin Tornquist “Técnicas de Redação, Conteúdo, Estilo e Testes de Requerimentos de Software”.

Apoiado pelo livro “The Requirements Engineering Handbook” de Ralph R. Young, Chapter 4 - Types of Requirements.

Apoiado pelo livro “Practical Software Requirements - A Manual of Content & Style”, de Benjamin L. Kovitz, Chapter 3 – Two worlds and three designs.

Apoiado pelo Texto “Implantação de um Processo de Gestão de Requisitos seguindo o CMMI” de Claudia Hazan, apresentado no VI Simpósio Internacional de Melhoria de Processos de Software.

Resumo Executivo

Diferentes literaturas, autores e escolas, fornecem diferentes definições e classificações para os diversos tipos de requisitos. A proposição de uma classificação é uma tarefa complexa, por ser complexa a própria engenharia de requisitos. Quanto mais tipos de requisitos apresentamos, mais aumentam a precisão da classificação. Contudo prejudicam o entendimento, tornando-a complexa demais para utilização. A sugestão é não gastar grande parte do tempo (e do custo) da organização definindo conceitos consensualmente, mas sim definindo o que é importante para construir e testar um produto que atenda as necessidades do usuário. Reforço que relevante é ter os requisitos necessários e suficientes para o entendimento dos participantes (*players*) do projeto.

Na prática, podemos utilizar conceitos simples para o dia-a-dia de uma organização em processo de amadurecimento e solidificação dos conceitos de engenharia de requisitos e de software. No texto que segue é sugerida uma classificação que tem como objetivo minimizar a complexidade do problema e facilitar sua utilização.

Conceituando Requisitos

Encontramos diversas definições e conceituações para requisitos. De forma simples, requisitos “definem o que é solicitado ao sistema fazer e com quais limitações ele é requisitado a operar”. Segundo o IEEE, requisito é uma “condição ou capacitação que deve ser contemplada pelo software, necessitada pelo usuário para resolver um problema ou alcançar um objetivo”. Segundo o Aurélio, requisito é (1) *uma condição que se deve satisfazer para que uma coisa fique legal e regular;* (2) *Exigência imprescindível para a consecução de certo fim;* (3) *Qualidade, dotes, predicados exigidos para um produto.*

De forma completa, “requisito é algo (verificável) que o produto deve fazer ou alguma qualidade (mensurável) que deve apresentar e que (pelo seu risco de comprometer o sucesso do projeto, compen\$\$a) deve ser testado”. Assim, os documentos que compõem uma especificação são essencialmente instrumentos contratuais, configurando um compromisso legal quanto ao escopo (O PORQUÊ), alcance (O QUÊ) e forma (O COMO) do produto a ser desenvolvido e testado.

Tipos de Requisitos

Tradicionalmente, os requisitos podem ser definidos em funcionais e não funcionais.

1. Os Requisitos Funcionais são as descrições das diversas operações que clientes e usuários querem (conhecidos também como requisitos de usuário), ou precisam que sejam realizadas pelo sistema.
2. Os Requisitos Não Funcionais são restrições ou atributos de qualidade de um software ou de um processo de desenvolvimento de software. É necessário que estes requisitos sejam considerados na fase inicial do processo de desenvolvimento de software. Recomendamos fortemente que os requisitos não funcionais sejam escritos na forma de Critérios de Aceite (vide *Test White Paper Uma Abordagem de Construção e Testes orientada pelos Critérios de Aceite* by Paulo Nannini).

Classificando os Requisitos Funcionais

Podemos agrupar os requisitos funcionais em três categorias diferentes, focando em resultados, descrevendo de forma integrada, do mais amplo ao mais específico:

1. **Requisitos de Negócio (RqN)** ou “*Business Why*” (1) fornecem a razão para o desenvolvimento de sistemas e software. (2) São derivados a partir dos objetivos de negócio. (3) É o que precisa ser entregue para agregar valor ao negócio do Cliente. (4) Focado no entendimento do “POR QUE” e “PARA QUE” o projeto/sistema é requerido

e, como contribuirá para os objetivos do negócio e do usuário. (5) Declaração escrita em linguagem de negócio contendo requisitos de alto-nível. (6) Devem ser escritos do ponto de vista do solicitante. (7) São a base para os requisitos de sistemas.

As seguintes perguntas devem ser respondidas:

- a) Porquê vamos gastar dinheiro com este projeto?
- b) O que o nosso negócio ganha com isso?
- c) O quê devemos entregar para gerar valor ao negócio?

Agora, tudo o que precisa ser entregue é “o quê o software faz” e “como ele faz” e, depende fortemente dos critérios de aceite do(s) requisito(s) de negócio para a decisão da melhor proposta de solução.

2. **Requisitos de Usuário (RqU)** ou “*Process What*” (1) descrevem “o quê” deve ser disponibilizado pelo processo/sistema/software para atender ao requisito de negócio. (2) Definem as funcionalidades esperadas sob o ponto de vista de funções. (3) Descrevem um comportamento que deve ser percebido pelo **usuário**. (4) Traduzem o que o sistema/produto precisa fazer para atender as necessidades dos usuários relacionados a um problema de negócio (5) Determinam “O QUE” é requerido (sem a preocupação do “COMO” será feito).
3. **Requisitos de Software (RqS)** ou “*System How*”, também encontrados como requisitos detalhados, técnicos, ou de implementação, (1) são derivados dos requisitos de usuário. (2) Focado no “COMO” os requisitos serão construídos e disponibilizados. (3) Escritos sob a ótica dos desenvolvedores.

Identificando outros componentes de uma especificação

1. Reescrevendo sobre Critérios de Aceite

Aproveitando as definições que utilizei no *Test White Paper “Uma Abordagem de Construção e Testes orientada pelos Critérios de Aceite”*, defino este como: uma frase que define o que deve ser verificado (presente/ausente) e/ou validado (funciona/não funciona) e/ou medido (acima, dentro dos limites, abaixo) do requisito especificado associado. Concordando James Robertson, critérios de aceite são objetivos mensuráveis, contendo números ou medidas, definindo o entendimento de um requisito

e verificando se a solução dada satisfaz a exigência original. A falta de um critério de aceite pode resultar em construção e/ou verificação inadequada, retrabalho custoso e/ou rejeição do cliente.

A vantagem em estabelecer critérios de aceite é que estes fornecem algum alvo quantificável que, quando testado, revelará o grau com que a solução está em conformidade com seus requisitos.

2. Reconhecendo as Regras de Negócio

A regras de negócio especificam as particularidades das funcionalidades a serem desenvolvidas. Estas regras podem abranger diversos assuntos como suas políticas, interesses, objetivos, compromissos éticos e sociais, obrigações contratuais, decisões estratégicas, leis, regulamentações e restrições do negócio. Estas regras podem ainda ser traduzidas em critérios de aceite associados a um requisitos de negócio ou, mais provavelmente, aos requisitos do usuário.

Exemplo de regra de negócio:

“O cliente **não** pode retirar mais de R\$ 600,00 por dia de sua conta pelo caixa eletrônico”.

<i>Requisito de Usuário</i>	<i>Critério de Aceite</i>	<i>Tipo</i>
<i>Retirar Valores via Caixa Eletrônico</i>	<i><= R\$ 600,00</i>	<i>Positivo</i>

Atente que, para evitar ambigüidades, escrevi o requisito e seu critério de aceite na forma “positiva”, eliminando a negação existente na frase.

3. E os casos de teste?

É a definição de um conjunto específico de inputs de teste, condições de execução e resultados esperados, identificados com a finalidade de avaliar um determinado requisito e o critério de aceite especificado. Para cada critério de aceite definido é possível gerar no mínimo 1 caso de teste. A coleção de casos de testes vai determinar o escopo dos testes a serem executados.

Segue abaixo exemplo de geração de casos de testes:

<i>Requisito de Usuário</i>	<i>Critério de Aceite</i>	<i>Casos de Teste</i>	<i>Tipo</i>
<i>Retirar Valores via Caixa Eletrônico</i>	$\leq R\$ 600,00$	$< R\$600,00$	<i>Positivo</i>
		$= R\$600,00$	<i>Positivo</i>
		$> R\$600,00$	<i>Negativo</i>

No exemplo acima, somente foi utilizada a técnica de *Particionamento por Equivalência* para a geração dos casos de testes, com a finalidade de ilustrar a criação de casos de testes a partir do critério de aceite especificado, podendo ser complementado ainda utilizando outras técnicas como, por exemplo, Análise de Valores Limite.

4. *Sugerindo uma solução para a problemática do requisito inverso (ou o “fora de escopo”)*

Este requisito quando especificado quer evidenciar o que o sistema **não pode** fazer, para evitar comportamentos adversos, que gerem prejuízos ou que pelas regras de negócio não são válidos. Isto significa uma restrição para ou do sistema.

Por exemplo: “A funcionalidade “X” não será visualizada através do Safari, somente para o IE e Firefox.”

Este requisito, anteriormente chamado de requisito inverso, pode claramente ser especificado como critério de aceite. No exemplo acima, é possível gerar 2 critérios de aceite positivos e 1 critério de aceite negativo:

<i>Requisito de Usuário</i>	<i>Critério de Aceite</i>	<i>Tipo</i>
<i>Visualizar Funcionalidade “X”</i>	<i>Visível no IE</i>	<i>Positivo</i>
	<i>Visível no Firefox</i>	<i>Positivo</i>
	<i>Visível no Safari</i>	<i>Negativo</i>

Entretanto, devo deixar claro que somente explicito este critério de aceite negativo se for uma solicitação do usuário, ou se for necessário evidenciar que o sistema funciona conforme solicitado ou *não faz o que não deveria fazer*.

Figurograma da Taxonomia de Requisitos

Abaixo apresentando um figurograma que tem como objetivo evidenciar a hierarquia entre os requisitos e critérios de aceite e mostrar quais as possíveis origens destes critérios de aceite.

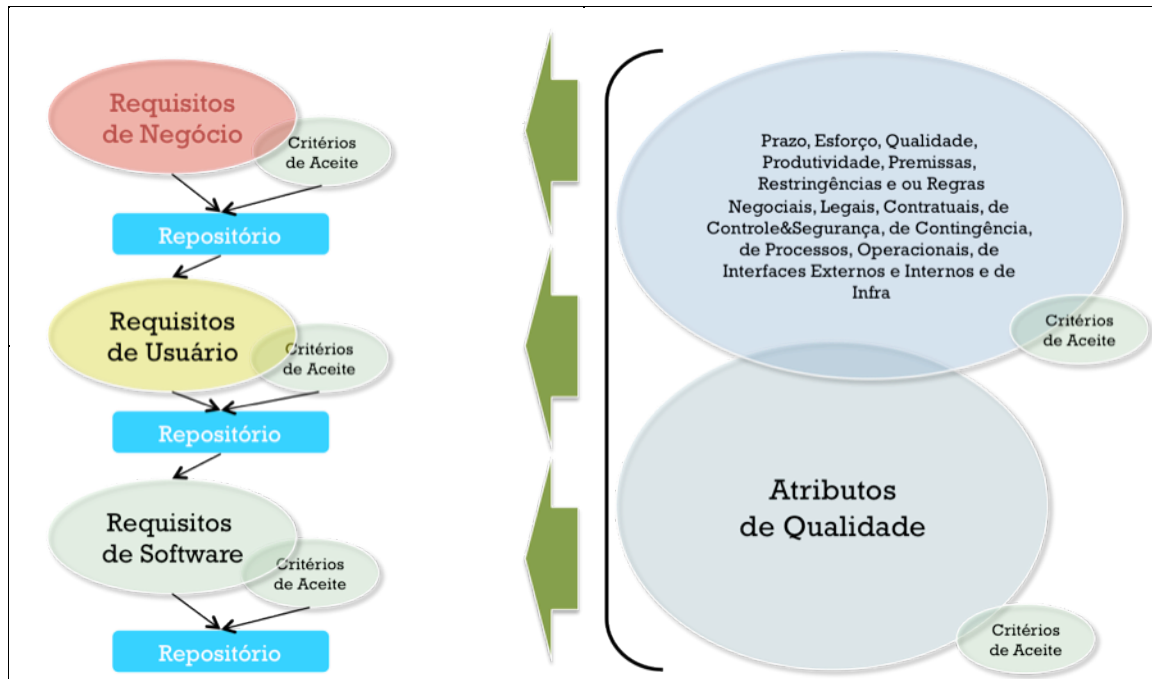


Figura 1 – Figurograma de Requisitos

Sobre o Autor

Paulo Henrique Nannini é Diretor da T&M Testes de Software e atual Presidente do Instituto Brasileiro de Qualidade em Testes de Software (IBQTS). Especialista em Gestão e Tecnologias da Qualidade, Escola Politécnica (MBA/USP). Atuou, em clientes como Banco Bradesco, Secretaria da Fazenda do Estado de São Paulo, Unisys do Brasil, BankBoston, Nextel, Boehringer, Nestlé, NetAge (Sênior Solution), DELL, ABN AMRO Bank, Banco Itaú, Orbitall, DATASUL (Totvs), REDECARD, Cardif, British American Tobacco, Banco HSBC, Serasa e BVMF.

Para maiores informações visite <http://web.me.com/paulonannini>