

[◀ VOLTAR](#)

CMMI - Área de Processo: Gestão de Requisitos

Detalhar a área de processo do CMMI-DEV nível 2 de gestão de requisitos.

NESTE TÓPICO

- › Requisito de Software - Definição
- › Requisitos Funcionais
- › Requisitos não Funcionais
- › Gestão de Requisitos



Requisito de Software - Definição

O Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE), em seu glossário de Terminologias para Engenharia de Software, define requisito assim:

Uma condição ou capacidade necessária para um usuário resolver um problema ou alcançar um objetivo.

((IEEE STANDARDS ASSOCIATION, 1990))

Os requisitos são as características que um software deve ter para atender às solicitações do cliente. Geralmente, os requisitos são divididos em dois tipos: funcionais e não funcionais.

Requisitos Funcionais

Os requisitos funcionais são aqueles que descrevem as funcionalidades que o sistema deve disponibilizar aos usuários. Dizem respeito às regras de negócio e, segundo Koscianki e Soares (2007), esses requisitos "devem estar desvinculados, o máximo possível, da tecnologia de implementação", ou seja, não devem ter uma dependência em relação ao tipo de linguagem, bancos de dados ou a qualquer outro elemento de configuração.

SÃO EXEMPLOS DE REQUISITOS FUNCIONAIS:

- Manter dados do cliente.
- Emitir relatório de vendas.
- Efetuar o cálculo da folha de pagamento.

Requisitos não Funcionais

Os requisitos não funcionais especificam ou restringem o comportamento do software, como por exemplo, a confiabilidade (nível de desempenho), o ambiente de desenvolvimento ou requisitos legais.

Esses requisitos não fazem parte das funcionalidades que serão solicitadas pelo cliente, mas são características mínimas que o software deverá cumprir.

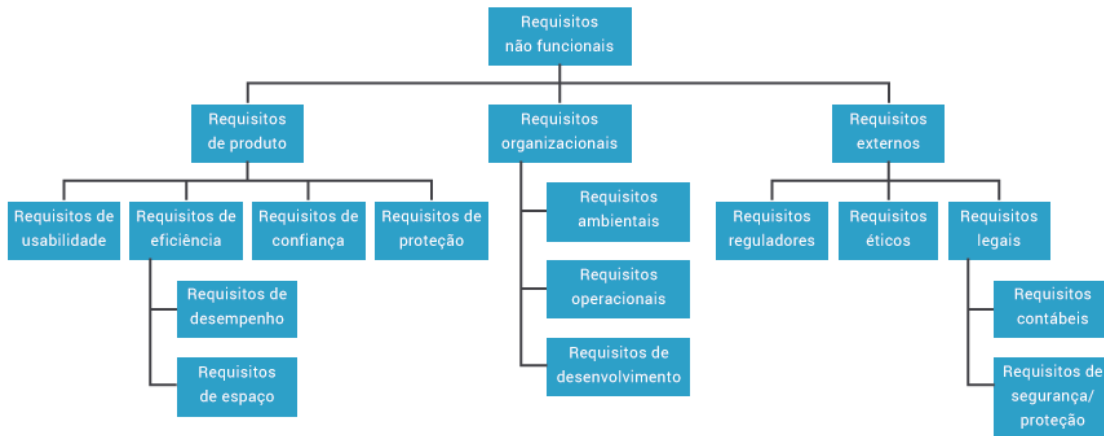
SÃO EXEMPLOS DE REQUISITOS NÃO FUNCIONAIS:

- Requisitos de Eficiência, como tempo de resposta.
- Requisitos impostos pela organização, como padrões para desenvolvimento ou a infraestrutura a ser utilizada.
- Requisitos de integração com outros sistemas



Observe que, pelos exemplos acima, esses requisitos não serão levantados junto aos usuários do sistema, eles serão observados pelo desenvolvedor por meio de outras fontes.

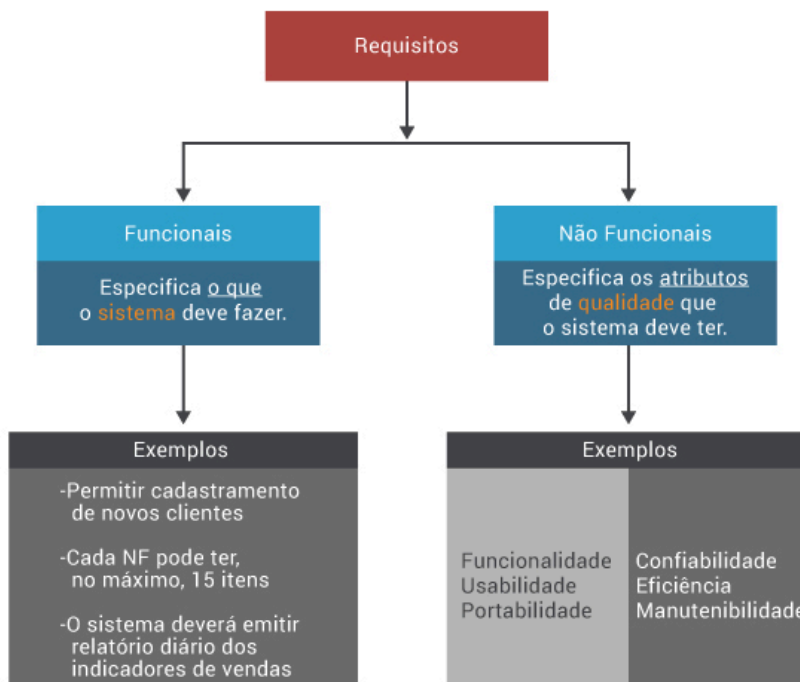
Algumas das principais fontes de requisitos não funcionais podem ser observadas na figura a seguir:



Fontes para Requisitos não Funcionais

A figura a seguir ilustra a diferenciação entre requisitos funcionais e não funcionais:

Tipos de Requisitos de Software



Tipos de Requisitos de Software: Funcionais e Não Funcionais

Fonte:

Gestão de Requisitos

Sommerville (2007) define a gestão de requisitos como "um processo para se compreender e controlar as mudanças de requisitos". Isso porque, em quase todo projeto de desenvolvimento e manutenção de sistemas, os requisitos de

software mudarão com certa frequência. Entre outros fatores, essas mudanças ocorrem em decorrência de:

- No início do projeto, muitas vezes, o problema não pode ser totalmente definido, tornando dessa forma o requisito ainda incompleto, onde mais tarde terá que ser atualizado;
- As partes interessadas, que definem os requisitos, mudam de entendimento sobre suas necessidades conforme o projeto do sistema vai progredindo;
- Quando o sistema já está instalado, rodando em produção, inevitavelmente novos requisitos surgirão.

Gerenciamento de Requisitos é um processo para compreender e controlar as mudanças dos requisitos do sistema, proporcionando um acompanhamento dos requisitos individualmente, além de manter as ligações entre os requisitos dependentes.

Os requisitos para sistemas baseados em computadores mudam, e o desejo de mudar os requisitos persiste ao longo da vida de um sistema. Gestão de requisitos é um conjunto de atividades que ajuda a equipe de projeto a identificar, controlar e acompanhar as necessidades e suas mudanças a qualquer momento enquanto o projeto prossegue.

((PRESSMAN, 2011))

Pode-se ainda acrescentar que a gestão de requisitos estabelece um acordo entre o cliente e a área de desenvolvimento de sistemas, em relação tanto à correta identificação dos requisitos dos sistemas, como em relação às alterações que esses requisitos sofrerão. Além disso, procura assegurar que os corretos serão utilizados para a adequada solução do problema e que o sistema desenvolvido será composto somente pelos elementos necessários.



Objetivos da área de gestão de requisitos

Essa área de processo do CMMI-DEV fornece subsídios para gerenciar os requisitos dos produtos e componentes de produto do projeto e identificar inconsistências entre esses requisitos e os planos e produtos de trabalho do projeto (SOFTWARE ENGINEERING INSTITUTE, 2010).

Os principais objetivos dessa área de processo são:

- Fornecer subsídios para gerenciar os requisitos dos produtos e componentes de produto do projeto;
- Obter Entendimento do Requisito, trabalhando junto aos provedores de requisitos para obtenção de um melhor entendimento do significado dos requisitos;
- Obter comprometimento dos participantes do projeto com os requisitos;
- Gerenciar mudanças nos requisitos à medida que evoluem durante o projeto;
- Manter rastreabilidade de requisitos, desde o cliente até o produto ou o componente de produto;

- Identificar inconsistências entre Produtos de Trabalho, Planos de Projeto e Requisitos.

Motivações para implantação da gestão de requisitos

O vídeo abaixo ilustra com uma divertida animação os problemas que ocorrem quando os requisitos não são bem compreendidos e administrados em um projeto.

./videos/370366259.mp4



Os problemas encontrados por desenvolvedores para a produção de sistemas são, muitas vezes, em razão da falha de comunicação entre ele e o cliente. Os requisitos, quando não estão claros, podem gerar interpretação deficiente, criando um produto que, na maioria das vezes, não era o que o consumidor tinha em mente. Além disso, é comum que a definição de um sistema sofra interferência de diversas áreas, cada uma contribuindo com ideias que podem acabar modificando os conceitos inicialmente estabelecidos.

As principais falhas de projetos de software são ocasionadas pela coleta, interpretação, aplicação ou gestão incorreta dos requisitos. Deve-se ter em mente que os requisitos funcionais são mutáveis, tanto durante o desenvolvimento do software quanto após a entrega e, por esse motivo, devem ser gerenciados, ou seja, existe a necessidade de controlar essas alterações, verificando sua origem, validade, necessidade, propósito e consistência.

Os principais objetivos da gestão de requisitos são os de garantir que:

- As mudanças nos requisitos sejam plenamente compreendidas.
- Os impactos decorrentes das mudanças de um requisito estejam corretamente dimensionados.
- As alterações nos requisitos estejam de acordo com as regras de negócio e legislação vigentes.

- As alterações de um requisito sejam identificadas e determinem os motivos, ou seja, deve estar claro quem solicitou a mudança e por qual motivo essa modificação foi solicitada.
- O custo x benefício da alteração seja adequadamente estimado.

A gerência de requisitos é um processo em que a compreensão sobre as mudanças deve ser o foco principal. Planejar o gerenciamento é imprescindível para que esse processo seja efetuado de forma correta. Algumas atividades devem ser consideradas para que exista efetivamente um controle dos requisitos de sistema:

- Identificação dos requisitos: deve ser uma identificação única para cada requisito.
- Processo de gerenciamento de mudanças: análise da mudança, custo e sua implementação fazem parte dessa atividade.
- Rastreabilidade: origem e relacionamentos que um requisito pode ter.
- Aquisição de ferramentas de apoio: a grande quantidade de informações sobre os requisitos deve ser mantida e manuseada por meio de ferramentas próprias para isso. Existem no mercado diversas opções, tanto comerciais como *open source*. Essas ferramentas são úteis para armazenar, gerenciar as mudanças e permitir a rastreabilidade dos requisitos.

Quiz

Exercício Final



CMMI - Área de Processo: Gestão de Requisitos

INICIAR ➤

Referências

BARTIÉ, A. *Garantia da qualidade de software*. Rio de Janeiro: Campus, 2002.

IEEE STANDARDS ASSOCIATION. *Standard Glossary of Software Engineering Terminology*. [S.l.]: Std 610.12-1990, 1990.

KOSCIANSKI, A.; SOARES, M. D. S. *Qualidade de software*: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software. 2. ed. São Paulo: Novatec, 2007.

PAULA FILHO, W. D. P. *Engenharia de Software*: fundamentos, métodos e padrões. Rio de Janeiro: LTC, 2003.

PFLEEGER, S. L. *Engenharia de Software*: teoria e prática. 2. ed. São Paulo: Pearson, 2004.

PRESSMAN, R. S. *Engenharia de Software*: Uma Abordagem Profissional. 7. ed. Porto Alegre: McGraw Hill, 2011.

SOFTWARE ENGINEERING INSTITUTE. *CMMI for Services Version 1.3*. Bedford: Carnegie Mellon, 2010. Disponível em: <www.sei.cmu.edu/reports/10tr033.pdf>. Acesso em: 21 set. 2016.

SOMMERVILLE, I. *Engenharia de Software*. 8. ed. São Paulo: Pearson Addison-Wesley, 2007.



Avalie este tópico



ANTERIOR

CMMI-DEV Nível 2 - Áreas de Processos

Biblioteca

(<https://www.uninove.br/conhec>-

a-

[uninove/biblioteca/sobre-](https://www.uninove.br/biblioteca/sobre)

a-

[biblioteca/apresentacao/](https://www.uninove.br/biblioteca/apresentacao/))

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site



Índice

CMMI - Áreas de Processos: Planejamento

Monitoramento e Controle de Proj

© Todos os direitos reservados

Ajuda?
([https://ava.un](https://ava.uninove.br/ava/curso/)
idCurso=)

