

[www.devmedia.com.br](http://www.devmedia.com.br)

[versão para impressão]

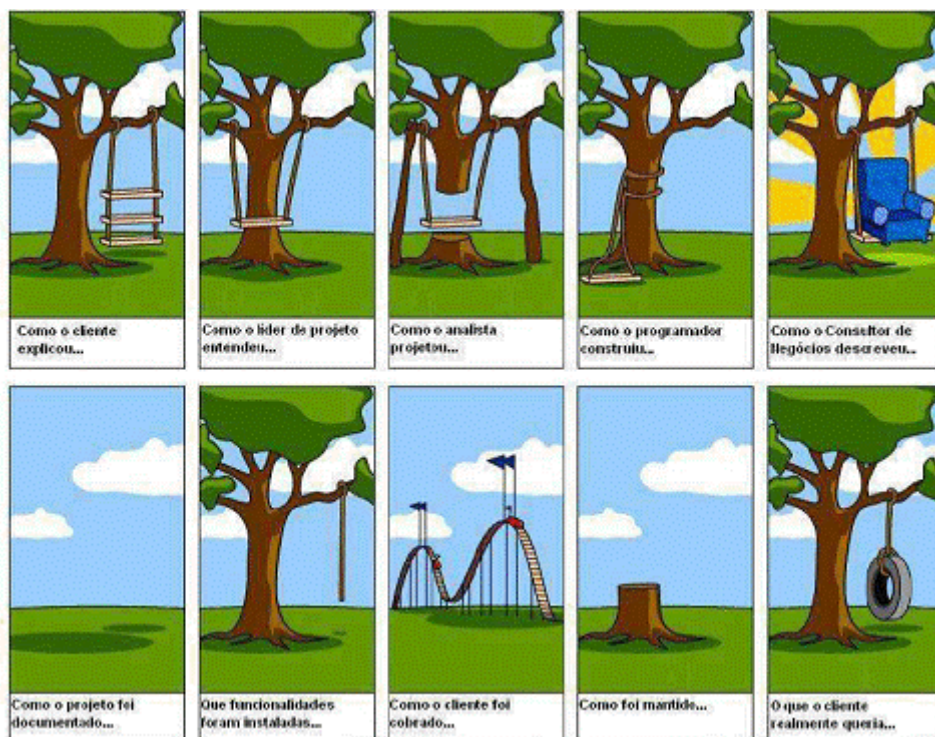
Link original: <https://www.devmedia.com.br/introducao-a-requisitos-de-software/29580>

## Introdução a Requisitos de Software

**Veja neste artigo o que são requisitos de software, quais são as suas classificações e como os requisitos estão relacionados com o ciclo de vida de um projeto de desenvolvimento de software.**

As falhas em requisitos estão entre as principais razões para o fracasso de um software. Entre as principais razões destacam-se os requisitos mal organizados, requisitos mal expressos, requisitos desnecessários para os clientes e a dificuldade para lidar com requisitos frequentemente mutáveis.

Abaixo temos uma figura clássica na área de engenharia de software (**Figura 1**) que demonstra como os requisitos podem significar um grande problema na especificação de um software.



**Figura 1.** Especificação de um software desde o cliente

Entre os problemas típicos encontrados com a especificação dos requisitos temos:

- Falta de conhecimento sobre o domínio: Isso normalmente leva a entendimentos errados ou mal compreendidos. Além disso, quando temos poucos ou nenhum modelo e anotações sofremos com esquecimentos no futuro, inclusive com alguns conceitos iniciais ou mais básicos que quando não tomamos nota ou modelamos temos mais chances de esquecer.

- **Comunicação do Usuário com o Analista:** Quando temos diferentes pontos de vista entre o Usuário e o Analista podemos ter requisitos mal feitos ou falhos. O interessante nesse caso é fazermos um Vocabulário da Aplicação anotando conceitos importantes e procurando fazer um vocabulário comum junto ao cliente.
- **Gestão de Mudanças e Evolução dos requisitos:** Um grande problema comumente encontrado é a falta de rastreabilidade ou de histórico de decisões de uma aplicação. Quanto mais a aplicação vai crescendo mais difícil fica de sabermos que pontos do sistema impactam em quais requisitos e também quais requisitos impactam em quais operações do sistema. Dessa forma é sempre importante termos uma rastreabilidade sobre quais impactos temos entre sistema e requisitos. Outra rastreabilidade existente é a chamada Rastreabilidade de Origem que diz quem definiu determinado requisito. Assim, quando precisamos saber quem solicitou determinada funcionalidade podemos saber de imediato o nome do cliente. Isso muitas vezes é importante até mesmo quando precisamos solicitar maiores detalhes para o cliente ou então fazer alguma negociação.

No restante do artigo veremos mais especificamente o que são requisitos, como eles são classificados e como os requisitos são utilizados durante todo o ciclo de vida de um projeto de desenvolvimento de software.

## Requisitos de Software

Antigamente dizia-se que requisitos eram sinônimos de funções, ou seja, tudo que o software deveria fazer funcionalmente. No entanto, atualmente assumiu-se que requisitos de software é muito mais do que apenas funções. Requisitos são, além de funções, objetivos, propriedades, restrições que o sistema deve possuir para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s). De forma mais geral um requisito é uma condição necessária para satisfazer um objetivo.

Portanto, um requisito é um aspecto que o sistema proposto deve fazer ou uma restrição no desenvolvimento do sistema. Vale ressaltar que em ambos os casos devemos sempre contribuir para resolver os problemas do cliente e não o que o programador ou um arquiteto deseja. Dessa forma, o conjunto dos requisitos como um todo representa um acordo negociado entre todas as partes interessadas no sistema. Isso também não significa que o programador, arquiteto ou um analista bem entendido no assunto de tecnologia não possam contribuir com sugestões e propostas que levem em conta o desejo do cliente.

Além disso, ainda temos um documento de requisitos que é uma coleção dos requisitos.

Por fim, os requisitos possuem alguns objetivos centrais como estabelecer e manter uma concordância com os clientes e outros envolvidos sobre o que o sistema deve fazer, deve oferecer aos desenvolvedores, projetistas e testadores do sistema uma compreensão melhor dos requisitos do sistema, definir fronteiras do sistema definindo o que deve ser incluído e o que não deve fazer parte do sistema, fornecer uma base para estimar o custo e o tempo de desenvolvimento do sistema e por fim definir uma interface de usuário para o sistema.

## Classificação dos Requisitos

Existem dois tipos de classificação de requisitos, são eles: Requisitos Funcionais (RF) e Requisitos Não-Funcionais (RNF).

Os requisitos funcionais referem-se sobre o que o sistema deve fazer, ou seja, suas funções e informações. Os requisitos não funcionais referem-se aos critérios que qualificam os requisitos funcionais. Esses critérios podem ser de qualidade para o software, ou seja, os requisitos de performance, usabilidade, confiabilidade, robustez, etc. Ou então, os critérios podem ser quanto a qualidade para o processo de software, ou seja, requisitos de entrega, implementação, etc.

Evidentemente que as prioridades podem variar conforme a natureza do software. Isso quer dizer que um software para uma plataforma de celular terá diferentes requisitos de um software que roda num browser na web. Assim como um software de tempo real que precisa ser executado em 1 segundo é diferente de outro software que pode ter um tempo maior para execução de uma determinada tarefa.

Portanto, requisitos funcionais preocupam-se com a funcionalidade e os serviços do sistema, ou seja, as funções que o sistema deve fornecer para o cliente e como o sistema se comportará em determinadas situações. Segue abaixo alguns exemplos de requisitos funcionais:

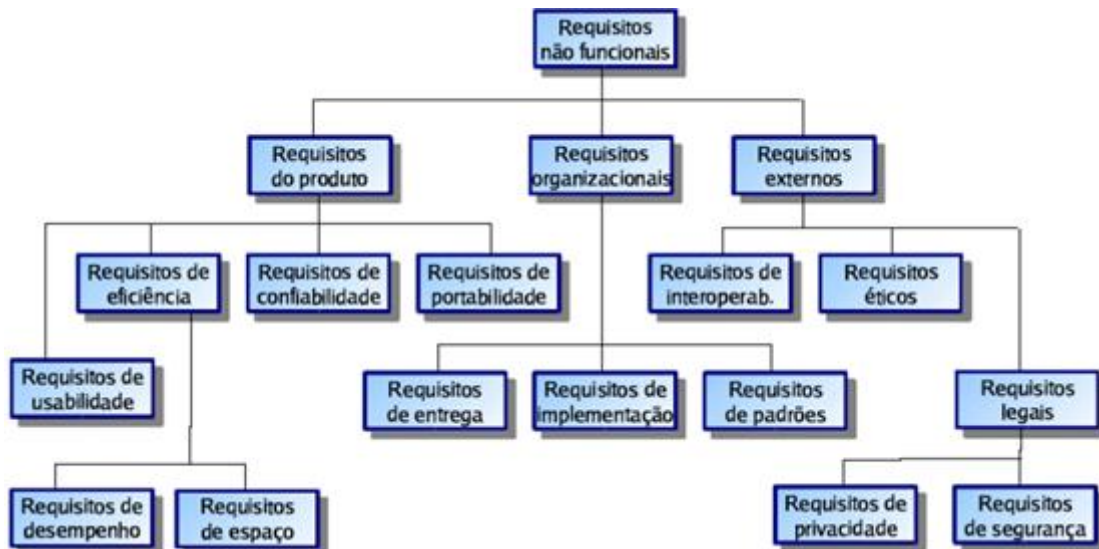
- [RF001] O Sistema deve cadastrar médicos profissionais (entrada)
- [RF002] O Sistema deve emitir um relatório de clientes (saída)
- [RF003] O Sistema deve passar um cliente da situação "em consulta" para "consultado" quando o cliente terminar de ser atendido (mudança de estado)
- [RF004] O cliente pode consultar seus dados no sistema

Por fim, os requisitos não funcionais definem propriedades e restrições do sistema como tempo, espaço, linguagens de programação, versões do compilador, SGBD, Sistema Operacional, método de desenvolvimento, etc. Uma dica importante é que os requisitos não funcionais são geralmente mensuráveis e assim devemos preferencialmente associar uma medida ou referência para cada requisito não funcional. A **Tabela 1** mostra diversas propriedades e métricas para cada uma delas.

Propriedade	Métrica
Velocidade	Transações processadas por segundo. Tempo de resposta ao usuário/evento. Tempo de atualização da tela.
Tamanho	Kbytes. Número de chips de RAM.
Facilidade de uso	Tempo de treinamento. Número de telas de ajuda.
Confiabilidade	Tempo médio para falhar. Probabilidade de indisponibilidade. Taxa de ocorrência de falhas. Disponibilidade.
Robustez	Tempo de reinício após falha. Porcentagem de eventos que causam falhas. Probabilidade de que os dados sejam corrompidos por falhas.
Portabilidade	Porcentagem de declarações dependentes de sistema-alvo. Número de sistemas-alvo.

**Tabela 1.** Propriedades e métricas para requisitos não funcionais

Os requisitos não funcionais ainda são classificados em três tipos, são eles: Requisitos do Produto Final, Requisitos Organizacionais e Requisitos Externos. Requisitos do Produto Final referem-se a como o produto deve comportar-se, ou seja, a sua velocidade de execução, confiabilidade, etc. Requisitos Organizacionais referem-se à consequência de políticas e procedimentos organizacionais que devem ser seguidos. Requisitos Externos referem-se a fatores externos ao sistema e ao processo de desenvolvimento como a legislação. Além desses três tipos, os requisitos não funcionais ainda se dividem em outros diversos tipos como demonstra a **Figura 2**.



**Figura 2.** Tipos de requisitos não funcionais

Segue abaixo alguns exemplos de requisitos não funcionais:

- [RNF001] O sistema deve imprimir o relatório em até 5 segundos.
- [RNF002] Todos os relatórios devem seguir o padrão de relatórios especificado pelo setor XYZ.
- [RNF003] O sistema deve ser implementado em Java.

Para ajudar os analistas a identificar o real propósito das informações obtidas junto aos usuários, existe um sistema de classificação de requisitos chamada FURPS+ que são as iniciais dos nomes abaixo, onde cada um tem um propósito:

- Functionality (Funcionalidade): É todo o aspecto funcional do sistema sendo desenvolvido.
- Usability (Usabilidade): Indica o tempo de treinamento para um usuário se tornar produtivo, Tempo de duração desejado para determinada operação no sistema e Ajuda on-line, documentação do usuário e material de treinamento.
- Reliability (Confiabilidade): Refere-se à Disponibilidade, Tempo de correção – tempo permitido para indisponibilidade quando ocorre uma falha, Precisão, Número máximo de defeitos (bugs/KLOC – mil linhas de código), Categorias de bugs – bugs devem ser categorizados por nível de impacto.
- Performance (Performace ou Desempenho): Indica o Tempo de resposta para uma transação, Troughput (ex: transações por segundos), Capacidade (ex: transações concorrentes), Operação Parcial (Situação do sistema aceitável quando estiver prejudicado de alguma forma), Uso de recursos: memória, espaço em disco, comunicação, etc.
- Supportability (Suportabilidade): Indica o Padrão de codificação, Convenção de nomenclatura, Bibliotecas de classes, Utilitários de manutenção.

- Plus (+): Indica Outros como Design, implementação, interface, físicos, etc.

## Requisitos no Ciclo de Vida do Projeto

Muitos pensam que os requisitos só devem ser considerados no início do projeto, o que não é verdade, pois os requisitos devem ser considerados durante todo o ciclo de vida de desenvolvimento do software.

Segue abaixo as etapas onde os requisitos são utilizados durante o ciclo de vida do projeto:

- Definição de critérios de aceitação e validação (pelos stakeholders);
- Definição sobre O QUÊ o sistema deve fazer (especificação para a equipe);
- Teste e Verificação do sistema sendo desenvolvido;
- Informações para gerenciamento de mudanças (Rastreabilidade, análise de impacto);
- Alocação de tarefas para a equipe;
- Estimativa de custo/esforço/cronograma;
- Acompanhamento e Controle do andamento do projeto.

Com isso, neste artigo vimos que falhas nos requisitos estão entre as mais importantes causas de fracasso dos sistemas de softwares. Os requisitos são definidos e especificados após a análise do sistema e a delimitação do escopo do software. Também vimos que os requisitos podem ser funcionais e não funcionais e que os requisitos fazem parte de todo o ciclo de vida de um projeto de desenvolvimento de software. Por fim a análise e a especificação dos requisitos de um software são atividades que determinam os objetivos de um software e as restrições associadas, bem como a elaboração da especificação do software. Ambas as atividades de análise e especificação são atividades interdependentes e devem ser realizadas conjuntamente.

### Bibliografia:

- [1] Sommerville, I. Engenharia de Software - 8ª Edição 2007.
- [2] Pressman, R. Software Engineering: A Practitioner's Approach. Makron Books, 2009.