

# Engenharia de Requisitos

Douglas O. Cardoso  
douglas.cardoso@cefet-rj.br



# Objetivos da aula

- Definir o conceito de requisito de software
- Diferenciar requisitos funcionais e não-funcionais
- Apresentar e relacionar as principais atividades de engenharia de requisitos

# Objetivos da aula

- Definir o conceito de requisito de software
- Diferenciar requisitos funcionais e não-funcionais
- Apresentar e relacionar as principais atividades de engenharia de requisitos

# Objetivos da aula

- Definir o conceito de requisito de software
- Diferenciar requisitos funcionais e não-funcionais
- Apresentar e relacionar as principais atividades de engenharia de requisitos

# Roteiro

## 1 Introdução

## 2 Requisitos funcionais e não-funcionais

## 3 Atividades básicas de engenharia de requisitos

# O que são requisitos de software?

Informalmente, indicações do que o software deve fazer

- Funções a realizar
- Serviços a oferecer
- Restrições na sua operação

# O que são requisitos de software?

Informalmente, indicações do que o software deve fazer

- Funções a realizar
- Serviços a oferecer
- Restrições na sua operação

# O que são requisitos de software?

Informalmente, indicações do que o software deve fazer

- Funções a realizar
- Serviços a oferecer
- Restrições na sua operação



# O que são requisitos de software?

Informalmente, indicações do que o software deve fazer

- Funções a realizar
- Serviços a oferecer
- Restrições na sua operação

# Por que requisitos de software são importantes?

- Servem de referência para o estabelecimento de contratos
- Garantem que o usuário, e não o desenvolvedor, guiam a criação do software
- Evitam discussões e confusão quanto a o que deve ser feito
  - ("O combinado não sai caro.")
- Ajuda a minimizar alterações que encarecem o desenvolvimento:
  - 3x mais barato que uma mudança durante o design
  - 5-10x mais barato uma mudança durante a implementação
  - 10-100 mais barato que uma mudança quando o software já está em uso

# Por que requisitos de software são importantes?

- Servem de referência para o estabelecimento de contratos
- Garantem que o usuário, e não o desenvolvedor, guiam a criação do software
- Evitam discussões e confusão quanto a o que deve ser feito
  - ("O combinado não sai caro.")
- Ajuda a minimizar alterações que encarecem o desenvolvimento:
  - 3x mais barato que uma mudança durante o design
  - 5-10x mais barato uma mudança durante a implementação
  - 10-100 mais barato que uma mudança quando o software já está em uso

# Por que requisitos de software são importantes?

- Servem de referência para o estabelecimento de contratos
- Garantem que o usuário, e não o desenvolvedor, guiam a criação do software
- Evitam discussões e confusão quanto a o que deve ser feito
  - (“O combinado não sai caro.”)
- Ajuda a minimizar alterações que encarecem o desenvolvimento:
  - 3x mais barato que uma mudança durante o design
  - 5-10x mais barato uma mudança durante a implementação
  - 10-100 mais barato que uma mudança quando o software já está em uso

# Por que requisitos de software são importantes?

- Servem de referência para o estabelecimento de contratos
- Garantem que o usuário, e não o desenvolvedor, guiam a criação do software
- Evitam discussões e confusão quanto a o que deve ser feito
  - (“O combinado não sai caro.”)
- Ajuda a minimizar alterações que encarecem o desenvolvimento:
  - 3x mais barato que uma mudança durante o design
  - 5-10x mais barato uma mudança durante a implementação
  - 10-100 mais barato que uma mudança quando o software já está em uso

# Por que requisitos de software são importantes?

- Servem de referência para o estabelecimento de contratos
- Garantem que o usuário, e não o desenvolvedor, guiam a criação do software
- Evitam discussões e confusão quanto a o que deve ser feito
  - (“O combinado não sai caro.”)
- Ajuda a minimizar alterações que encarecem o desenvolvimento:
  - 3x mais barato que uma mudança durante o design
  - 5-10x mais barato uma mudança durante a implementação
  - 10-100 mais barato que uma mudança quando o software já está em uso

# Por que requisitos de software são importantes?

- Servem de referência para o estabelecimento de contratos
- Garantem que o usuário, e não o desenvolvedor, guiam a criação do software
- Evitam discussões e confusão quanto a o que deve ser feito
  - (“O combinado não sai caro.”)
- Ajuda a minimizar alterações que encarecem o desenvolvimento:
  - 3x mais barato que uma mudança durante o design
  - 5-10x mais barato uma mudança durante a implementação
  - 10-100 mais barato que uma mudança quando o software já está em uso

# Por que requisitos de software são importantes?

- Servem de referência para o estabelecimento de contratos
- Garantem que o usuário, e não o desenvolvedor, guiam a criação do software
- Evitam discussões e confusão quanto a o que deve ser feito
  - (“O combinado não sai caro.”)
- Ajuda a minimizar alterações que encarecem o desenvolvimento:
  - 3x mais barato que uma mudança durante o design
  - 5-10x mais barato uma mudança durante a implementação
  - 10-100 mais barato que uma mudança quando o software já está em uso



# Por que requisitos de software são importantes?

- Servem de referência para o estabelecimento de contratos
- Garantem que o usuário, e não o desenvolvedor, guiam a criação do software
- Evitam discussões e confusão quanto a o que deve ser feito
  - (“O combinado não sai caro.”)
- Ajuda a minimizar alterações que encarecem o desenvolvimento:
  - 3x mais barato que uma mudança durante o design
  - 5-10x mais barato uma mudança durante a implementação
  - 10-100 mais barato que uma mudança quando o software já está em uso

# Níveis de abstração/detalhamento dos requisitos

- Comunicação adaptada aos interlocutores
- Requisitos de usuário: “gerar um relatório mensal sobre o estoque”
- Requisitos do sistema: conteúdo do relatório, restrições de acesso dos usuários ao relatório, data e hora da geração, etc

# Níveis de abstração/detalhamento dos requisitos

- Comunicação adaptada aos interlocutores
- Requisitos de usuário: “gerar um relatório mensal sobre o estoque”
- Requisitos do sistema: conteúdo do relatório, restrições de acesso dos usuários ao relatório, data e hora da geração, etc

# Níveis de abstração/detalhamento dos requisitos

- Comunicação adaptada aos interlocutores
- Requisitos de usuário: “gerar um relatório mensal sobre o estoque”
- Requisitos do sistema: conteúdo do relatório, restrições de acesso dos usuários ao relatório, data e hora da geração, etc

# Roteiro

1 Introdução

2 Requisitos funcionais e não-funcionais

3 Atividades básicas de engenharia de requisitos

# Requisitos funcionais

- Descrições de serviços e operações do sistema
- Suas entradas e respectivas saídas esperadas
- Reações a condições previstas

# Requisitos funcionais

- Descrições de serviços e operações do sistema
- Suas entradas e respectivas saídas esperadas
- Reações a condições previstas

# Requisitos funcionais

- Descrições de serviços e operações do sistema
- Suas entradas e respectivas saídas esperadas
- Reações a condições previstas



# Requisitos não-funcionais

- Restrições adicionais sobre funcionalidades do sistema
- Referentes a funcionalidades individuais (relacionadas a um requisito funcional)
- ... Ou a partes compostas ou o todo do sistema

# Requisitos não-funcionais

- Restrições adicionais sobre funcionalidades do sistema
- Referentes a funcionalidades individuais (relacionadas a um requisito funcional)
- ... Ou a partes compostas ou o todo do sistema

# Requisitos não-funcionais

- Restrições adicionais sobre funcionalidades do sistema
- Referentes a funcionalidades individuais (relacionadas a um requisito funcional)
- ... Ou a partes compostas ou o todo do sistema

# A importância de ambos os tipos de requisitos

- O software deve fazer o que se espera
- ... Mas também da forma como se espera
- Exemplo: você pegaria um avião cujo motor falha em 90% dos vôos?
- A classificação de um requisito como funcional ou não-funcional pode ser subjetiva
- Um requisito não-funcional (por exemplo, “restringir acesso a informações sensíveis”) pode provocar a definição de outros funcionais (definir contas de usuário e diretivas de acesso)

# A importância de ambos os tipos de requisitos

- O software deve fazer o que se espera
- ... Mas também da forma como se espera
- Exemplo: você pegaria um avião cujo motor falha em 90% dos vôos?
- A classificação de um requisito como funcional ou não-funcional pode ser subjetiva
- Um requisito não-funcional (por exemplo, “restringir acesso a informações sensíveis”) pode provocar a definição de outros funcionais (definir contas de usuário e diretivas de acesso)

# A importância de ambos os tipos de requisitos

- O software deve fazer o que se espera
- ... Mas também da forma como se espera
- Exemplo: você pegaria um avião cujo motor falha em 90% dos vôos?
- A classificação de um requisito como funcional ou não-funcional pode ser subjetiva
- Um requisito não-funcional (por exemplo, “restringir acesso a informações sensíveis”) pode provocar a definição de outros funcionais (definir contas de usuário e diretivas de acesso)

# A importância de ambos os tipos de requisitos

- O software deve fazer o que se espera
- ... Mas também da forma como se espera
- Exemplo: você pegaria um avião cujo motor falha em 90% dos vôos?
- A classificação de um requisito como funcional ou não-funcional pode ser subjetiva
- Um requisito não-funcional (por exemplo, “restringir acesso a informações sensíveis”) pode provocar a definição de outros funcionais (definir contas de usuário e diretivas de acesso)

# A importância de ambos os tipos de requisitos

- O software deve fazer o que se espera
- ... Mas também da forma como se espera
- Exemplo: você pegaria um avião cujo motor falha em 90% dos vôos?
- A classificação de um requisito como funcional ou não-funcional pode ser subjetiva
- Um requisito não-funcional (por exemplo, “restringir acesso a informações sensíveis”) pode provocar a definição de outros funcionais (definir contas de usuário e diretivas de acesso)



## Extra: requisitos do domínio

- Regras e restrições especificamente relacionadas ao domínio de aplicação do software
- Possivelmente óbvias para o cliente, mas não para os engenheiros de software
- “A quantidade de um item no estoque é sempre não-negativa”
- “O histórico de operações de cada conta bancária deve ser de pelo menos 6 meses”

## Extra: requisitos do domínio

- Regras e restrições especificamente relacionadas ao domínio de aplicação do software
- Possivelmente óbvias para o cliente, mas não para os engenheiros de software
- “A quantidade de um item no estoque é sempre não-negativa”
- “O histórico de operações de cada conta bancária deve ser de pelo menos 6 meses”

## Extra: requisitos do domínio

- Regras e restrições especificamente relacionadas ao domínio de aplicação do software
- Possivelmente óbvias para o cliente, mas não para os engenheiros de software
- “A quantidade de um item no estoque é sempre não-negativa”
- “O histórico de operações de cada conta bancária deve ser de pelo menos 6 meses”

## Extra: requisitos do domínio

- Regras e restrições especificamente relacionadas ao domínio de aplicação do software
- Possivelmente óbvias para o cliente, mas não para os engenheiros de software
- “A quantidade de um item no estoque é sempre não-negativa”
- “O histórico de operações de cada conta bancária deve ser de pelo menos 6 meses”

# Roteiro

1 Introdução

2 Requisitos funcionais e não-funcionais

3 Atividades básicas de engenharia de requisitos

# Estudo de viabilidade

- De forma resumida, avalia as seguintes perguntas:
- O sistema contribuirá para os objetivos gerais da organização?
- O desenvolvimento é possível dadas restrições tecnológicas, organizacionais, orçamentárias?
- Caso necessário, uma futura integração com outros sistemas seria possível?
- Em caso de resposta negativa para alguma pergunta, reconsiderar a continuidade do projeto.

# Estudo de viabilidade

- De forma resumida, avalia as seguintes perguntas:
- O sistema contribuirá para os objetivos gerais da organização?
- O desenvolvimento é possível dadas restrições tecnológicas, organizacionais, orçamentárias?
- Caso necessário, uma futura integração com outros sistemas seria possível?
- Em caso de resposta negativa para alguma pergunta, reconsiderar a continuidade do projeto.

# Estudo de viabilidade

- De forma resumida, avalia as seguintes perguntas:
- O sistema contribuirá para os objetivos gerais da organização?
- O desenvolvimento é possível dadas restrições tecnológicas, organizacionais, orçamentárias?
- Caso necessário, uma futura integração com outros sistemas seria possível?
- Em caso de resposta negativa para alguma pergunta, reconsiderar a continuidade do projeto.



# Estudo de viabilidade

- De forma resumida, avalia as seguintes perguntas:
- O sistema contribuirá para os objetivos gerais da organização?
- O desenvolvimento é possível dadas restrições tecnológicas, organizacionais, orçamentárias?
- Caso necessário, uma futura integração com outros sistemas seria possível?
- Em caso de resposta negativa para alguma pergunta, reconsiderar a continuidade do projeto.

# Estudo de viabilidade

- De forma resumida, avalia as seguintes perguntas:
- O sistema contribuirá para os objetivos gerais da organização?
- O desenvolvimento é possível dadas restrições tecnológicas, organizacionais, orçamentárias?
- Caso necessário, uma futura integração com outros sistemas seria possível?
- Em caso de resposta negativa para alguma pergunta, reconsiderar a continuidade do projeto.

# Elicitação (ou identificação) de requisitos

Atividade cíclica, iterativa e incremental através da qual requisitos são estabelecidos ou alterados. Em 4 passos:

- Descoberta
- Classificação e organização
- Priorização e negociação (solução de conflitos entre stakeholders)

# Elicitação (ou identificação) de requisitos

Atividade cíclica, iterativa e incremental através da qual requisitos são estabelecidos ou alterados. Em 4 passos:

- Descoberta
- Classificação e organização
- Priorização e negociação (solução de conflitos entre stakeholders)

# Elicitação (ou identificação) de requisitos

Atividade cíclica, iterativa e incremental através da qual requisitos são estabelecidos ou alterados. Em 4 passos:

- Descoberta
- Classificação e organização
- Priorização e negociação (solução de conflitos entre stakeholders)

# Elicitação (ou identificação) de requisitos

Atividade cíclica, iterativa e incremental através da qual requisitos são estabelecidos ou alterados. Em 4 passos:

- Descoberta
- Classificação e organização
- Priorização e negociação (solução de conflitos entre stakeholders)

# Elicitação: fontes de informação

- Stakeholders (clientes, usuários, especialistas no domínio etc)
- Documentos, manuais
- Legislação em vigor

# Elicitação: fontes de informação

- Stakeholders (clientes, usuários, especialistas no domínio etc)
- Documentos, manuais
- Legislação em vigor



# Elicitação: fontes de informação

- Stakeholders (clientes, usuários, especialistas no domínio etc)
- Documentos, manuais
- Legislação em vigor

# Elicitação: processos de coleta de informações

- Entrevistas
- Questionários
- Discussão de cenários e casos de uso
- Estudo etnográfico

# Elicitação: processos de coleta de informações

- Entrevistas
- Questionários
- Discussão de cenários e casos de uso
- Estudo etnográfico

# Elicitação: processos de coleta de informações

- Entrevistas
- Questionários
- Discussão de cenários e casos de uso
- Estudo etnográfico

# Elicitação: processos de coleta de informações

- Entrevistas
- Questionários
- Discussão de cenários e casos de uso
- Estudo etnográfico

# Documentação dos requisitos

- Geração da documentação formal que detalha cada requisito do software
- Tal documentação será usada para nortear todo o desenvolvimento
- Formato: texto, diagramas, fórmulas etc (mais detalhes em breve)

# Documentação dos requisitos

- Geração da documentação formal que detalha cada requisito do software
- Tal documentação será usada para nortear todo o desenvolvimento
- Formato: texto, diagramas, fórmulas etc (mais detalhes em breve)

# Documentação dos requisitos

- Geração da documentação formal que detalha cada requisito do software
- Tal documentação será usada para nortear todo o desenvolvimento
- Formato: texto, diagramas, fórmulas etc (mais detalhes em breve)



# Validação de requisitos

Checagem da equivalência entre os requisitos levantados e o que o cliente deseja. Isto inclui verificar:

- Validade (ou seja, os requisitos são razoáveis para o cliente)
- Consistência
- Completude (ou seja, tudo o que o cliente deseja foi coberto pelos requisitos)
- Realismo
- Verificabilidade

# Validação de requisitos

Checagem da equivalência entre os requisitos levantados e o que o cliente deseja. Isto inclui verificar:

- Validade (ou seja, os requisitos são razoáveis para o cliente)
- Consistência
- Completude (ou seja, tudo o que o cliente deseja foi coberto pelos requisitos)
- Realismo
- Verificabilidade

# Validação de requisitos

Checagem da equivalência entre os requisitos levantados e o que o cliente deseja. Isto inclui verificar:

- Validade (ou seja, os requisitos são razoáveis para o cliente)
- Consistência
- Completude (ou seja, tudo o que o cliente deseja foi coberto pelos requisitos)
- Realismo
- Verificabilidade

# Validação de requisitos

Checagem da equivalência entre os requisitos levantados e o que o cliente deseja. Isto inclui verificar:

- Validade (ou seja, os requisitos são razoáveis para o cliente)
- Consistência
- Completude (ou seja, tudo o que o cliente deseja foi coberto pelos requisitos)
- Realismo
- Verificabilidade

# Validação de requisitos

Checagem da equivalência entre os requisitos levantados e o que o cliente deseja. Isto inclui verificar:

- Validade (ou seja, os requisitos são razoáveis para o cliente)
- Consistência
- Completude (ou seja, tudo o que o cliente deseja foi coberto pelos requisitos)
- Realismo
- Verificabilidade

# Validação de requisitos

Checagem da equivalência entre os requisitos levantados e o que o cliente deseja. Isto inclui verificar:

- Validade (ou seja, os requisitos são razoáveis para o cliente)
- Consistência
- Completude (ou seja, tudo o que o cliente deseja foi coberto pelos requisitos)
- Realismo
- Verificabilidade

# Validação: algumas ferramentas

- Revisão sistemática
- Prototipação
- Geração de casos de teste

# Validação: algumas ferramentas

- Revisão sistemática
- Prototipação
- Geração de casos de teste



# Validação: algumas ferramentas

- Revisão sistemática
- Prototipação
- Geração de casos de teste

# Gestão de requisitos

Processo paralelo ao desenvolvimento, responsável pela manutenção dos requisitos e controle das alterações destes. Alguns pontos-chave desta atividade:

- Identificação unívoca de requisitos
- Rastreabilidade, relacionamentos entre requisitos
- Rotina de gerenciamento de mudanças
- Análise de impacto de alterações

# Gestão de requisitos

Processo paralelo ao desenvolvimento, responsável pela manutenção dos requisitos e controle das alterações destes. Alguns pontos-chave desta atividade:

- Identificação unívoca de requisitos
- Rastreabilidade, relacionamentos entre requisitos
- Rotina de gerenciamento de mudanças
- Análise de impacto de alterações

# Gestão de requisitos

Processo paralelo ao desenvolvimento, responsável pela manutenção dos requisitos e controle das alterações destes. Alguns pontos-chave desta atividade:

- Identificação unívoca de requisitos
- Rastreabilidade, relacionamentos entre requisitos
- Rotina de gerenciamento de mudanças
- Análise de impacto de alterações

# Gestão de requisitos

Processo paralelo ao desenvolvimento, responsável pela manutenção dos requisitos e controle das alterações destes. Alguns pontos-chave desta atividade:

- Identificação unívoca de requisitos
- Rastreabilidade, relacionamentos entre requisitos
- Rotina de gerenciamento de mudanças
- Análise de impacto de alterações

# Gestão de requisitos

Processo paralelo ao desenvolvimento, responsável pela manutenção dos requisitos e controle das alterações destes. Alguns pontos-chave desta atividade:

- Identificação unívoca de requisitos
- Rastreabilidade, relacionamentos entre requisitos
- Rotina de gerenciamento de mudanças
- Análise de impacto de alterações