

Engenharia de software

APRESENTAÇÃO

GISELE LOURENÇO

- Analista de produtos / Product Owner.
- Mestra em Gestão do Conhecimento (UNICESUMAR, 2020)
- Graduada em Engenharia de Software (UNICESUMAR, 2017)
- Certificada PSPO - PROFESSIONAL SCRUM PRODUCT OWNER™ | Scrum.org



[LINKEDIN](#) : GISELE CAROLINE URBANO LOURENÇO

Habilidades a serem desenvolvidas na disciplina

- Compreender os fundamentos da orientação a objeto para modelagem de software.
- Aprender a linguagem UML para realização da modelagem de software.
- Desenvolver capacidade analítica para levantamento de requisitos de software.
- Selecionar tipos adequados de diagramas para cada situação durante a modelagem de software.
- Contrastar os diferentes modelos de desenvolvimento de software de acordo com o projeto a ser desenvolvido.

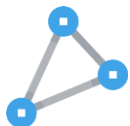
PLANO DE ESTUDOS

AULA 1



Título: Introdução à Engenharia de Software e metodologias ágeis

AULA 2



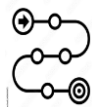
Título: Modelagem de sistemas

AULA 3



Título: Diagrama de sistemas

AULA 4



Título: Gerenciamento de Software

O QUE É SOFTWARE? QUAIS SUAS CARACTERÍSTICAS?



O software é desenvolvido ou passa por um processo de engenharia, não sendo fabricado no sentido clássico.



Software não se desgasta.



Embora a indústria caminhe para a construção com base em componentes, a maioria dos softwares continua a ser construída de forma personalizada (sob encomenda).



O termo Engenharia de Software foi utilizado a primeira vez em uma conferência da Organização do Tratado do Atlântico Norte (OTAN), conduzida na Alemanha em 1968 .

O que significa Engenharia de Software?



A engenharia de software é uma disciplina cujo foco está no desenvolvimento de sistemas de software de alta qualidade por um custo acessível (TSUI; KARAM, 2013).



Segundo Sommerville (2011, p. 4), “quando falamos de engenharia de software, não se trata apenas do programa em si, mas de toda a documentação associada e dados de configurações necessários para fazer este programa operar corretamente”.





Tipos de Softwares

- Software de sistema: (e.g. editores, compiladores, sistemas operacionais e softwares de rede).
- Software de aplicação: (e.g. ERP).
- Software de engenharia/científico: (e.g., software de vulcanologia).
- Software embarcado: (e.g., aparelhos celulares, micro-ondas);
- Software para linha de produtos: (e.g., softwares de linha de montagem de veículos);

Continuação...



Aplicações Web/aplicativos móveis: contemplam os softwares voltados a navegadores ou dispositivos móveis (e.g. site do mercado livre, aplicativo do mercado livre).



Software de inteligência artificial: utilizam algoritmos não numéricos para solucionar problemas complexos que não poderiam ser solucionados pela computação ou análise direta (e.g. marketing, assistentes virtuais);

A adaptação do software



O software deve ser adaptado para atender às necessidades de novos ambientes ou de novas tecnologias computacionais.



-O software deve ser aperfeiçoado para implementar novos requisitos de negócio.



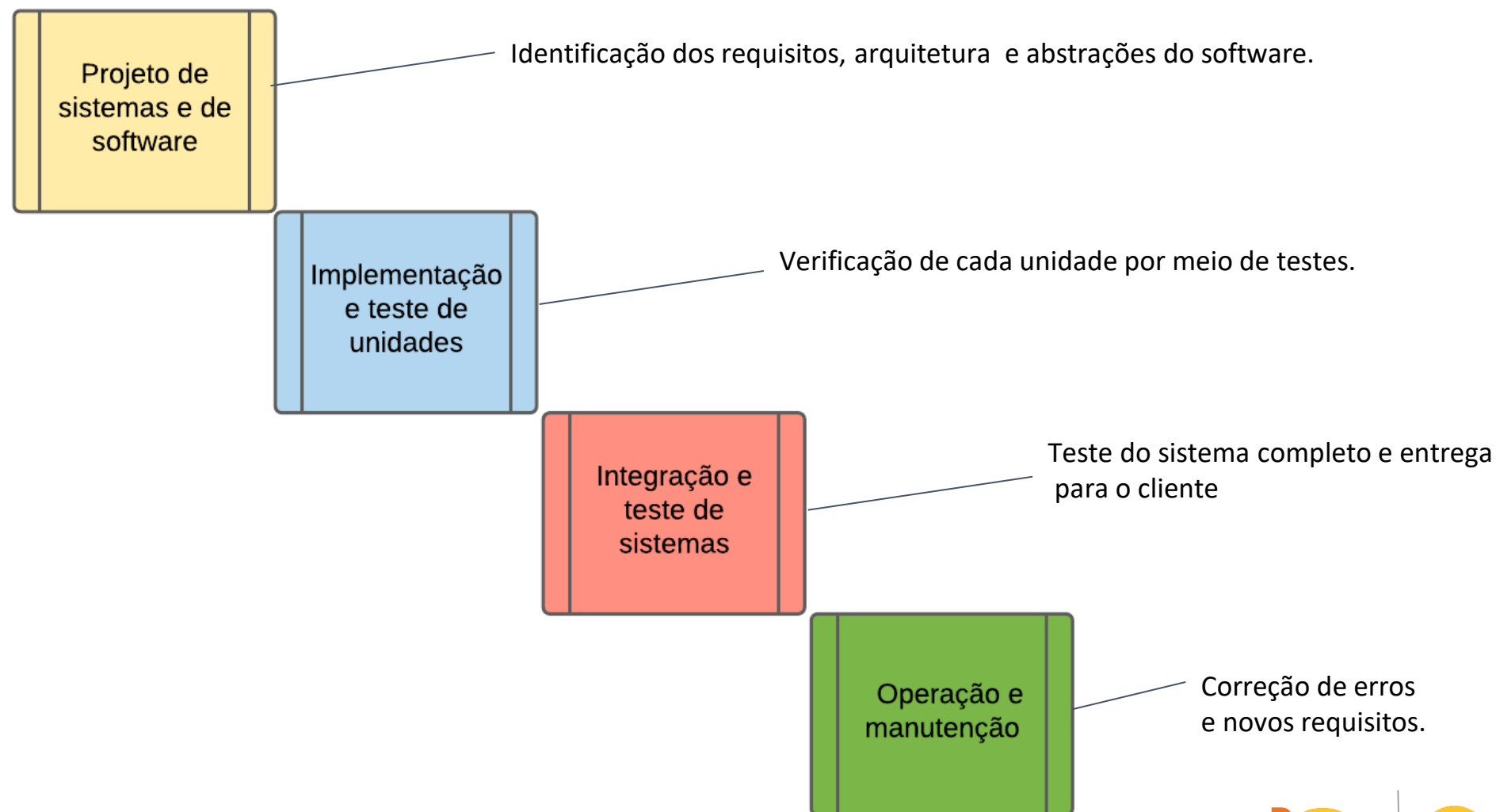
O software deve ser expandido para torná-lo capaz de funcionar com outros bancos de dados ou com sistemas mais modernos.



O software deve ser re-arquitetado para torná-lo viável dentro de um ambiente computacional em evolução (PRESSMAN; MAXIM, 2016, p. 8)

01	Especificação de software	<ul style="list-style-type: none">• É necessário que o cliente defina as funcionalidades do software que será desenvolvido, bem como todas as suas restrições operacionais.
02	Projeto e implementação de software	<ul style="list-style-type: none">• O software deve ser confeccionado mediante as especificações definidas anteriormente
03	Validação de software	<ul style="list-style-type: none">• O software precisa ser validado, a fim de garantir que ele faça o que o cliente deseja, ou seja, que atenda às especificações de funcionalidade.
04	Evolução de software	<ul style="list-style-type: none">• As funcionalidades definidas pelo cliente durante o desenvolvimento do software podem mudar. Consequentemente, o software precisará evoluir para atender a essas mudanças.

Modelo Cascata



Modelo cascata



Um estágio só pode ser iniciado depois que o anterior tenha sido concluído.



Na maioria das vezes, o cliente não consegue definir claramente todas as suas necessidades e o modelo em cascata requer essa definição no início das atividades.



Uma versão operacional do sistema somente estará disponível no final do projeto.

Então quando devo utilizar o cascata?



O modelo em cascata deve ser utilizado somente quando os requisitos são fixos e com pouca probabilidade de alteração.

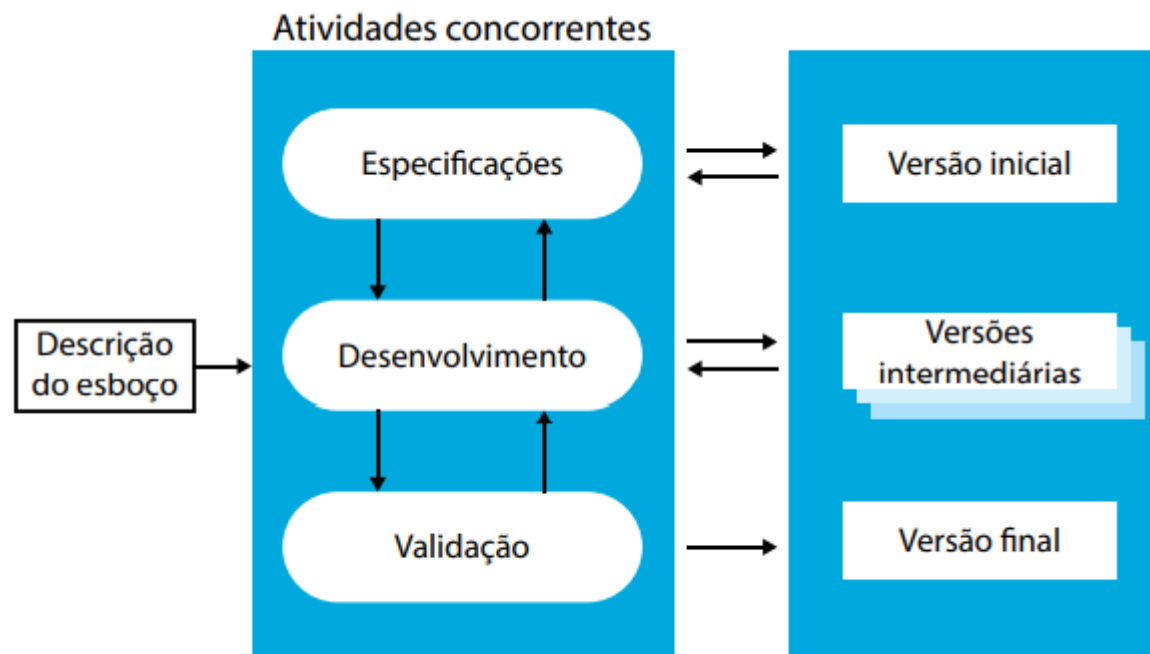


Deve ser considerado para utilização em softwares críticos, como por exemplo, sistema de controle aéreo.

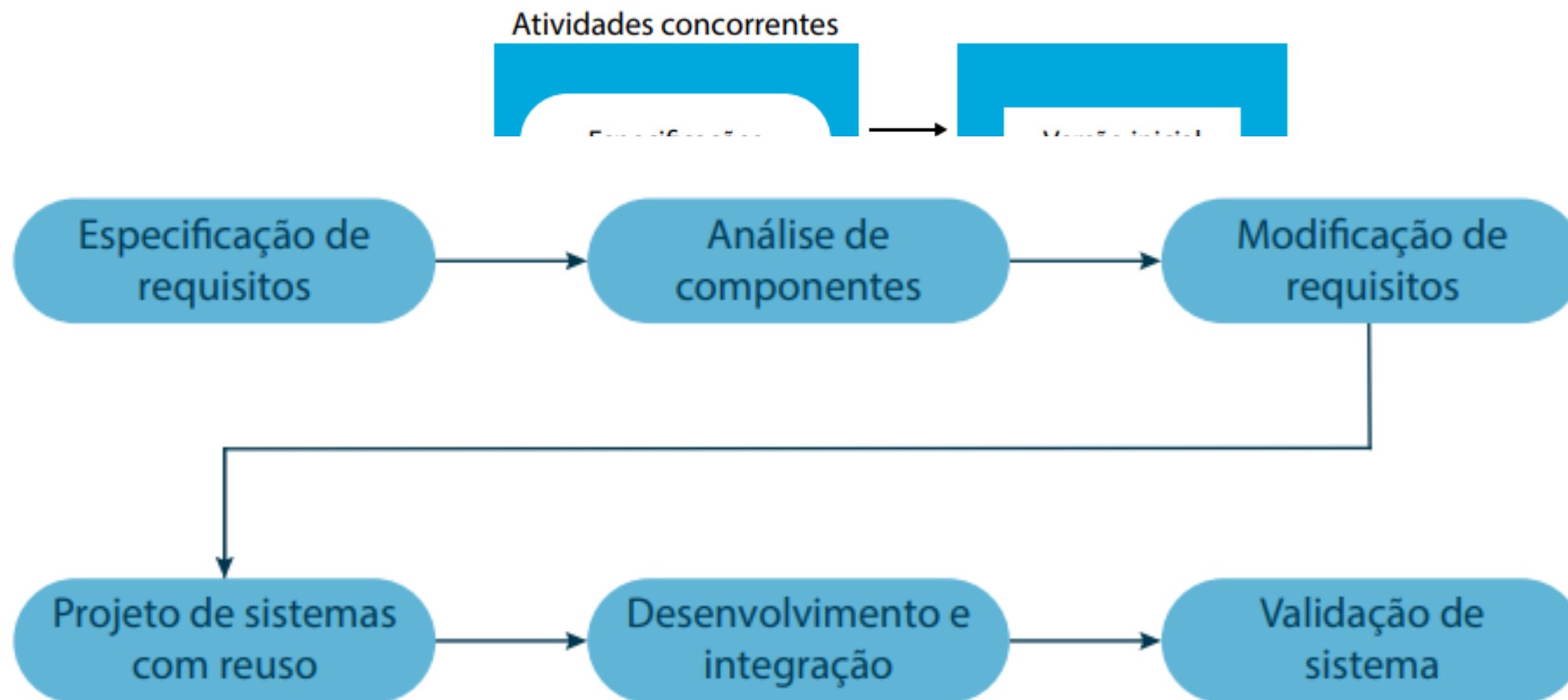


Softwares de curta duração de desenvolvimento.

ENGENHARIA DE SOFTWARE ORIENTADA A REUSO



ENGENHARIA DE SOFTWARE ORIENTADA A REUSO



ESPECIFICAÇÃO DE SOFTWARE

É a primeira atividade básica de um processo de software e tem, como objetivo, definir quais funções são requeridas pelo sistema.

1. Estudo de viabilidade: uma avaliação é realizada para verificar se as necessidades dos usuários.
2. Levantamento e análise de requisitos: são levantados os requisitos do sistema.
3. Especificação de requisitos: criação de um documento baseado na fase anterior.
4. Validação de requisitos: essa atividade verifica os requisitos quanto a sua pertinência, consistência e integralidade.



PROJETO E IMPLEMENTAÇÃO DE SOFTWARE

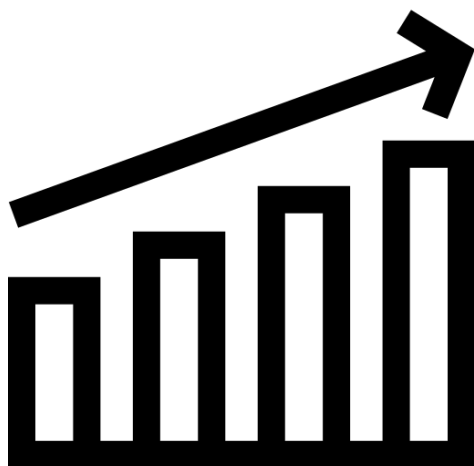


- i) Definição do diagrama geral do sistema;
- ii) Elaboração as interfaces com o usuário (telas e relatórios);
- iiii) Desenvolvimento de um conjunto de especificações de casos de uso. É necessário reforçar que essas especificações devem conter detalhes suficientes para permitir a sua codificação.

EVOLUÇÃO DE SOFTWARE

Depois que um software é colocado em funcionamento, ou seja, depois que é implantado, com certeza, ocorrerão mudanças que levarão à alteração desse software.

Os estágios de desenvolvimento e de manutenção têm sido considerados integrados e contínuos, em vez de dois processos separados.



VALIDAÇÃO DE SOFTWARE

A validação de software se dedica em mostrar que um sistema atende tanto as especificações relacionadas no documento de requisitos quanto às expectativas dos seus usuários. Sommerville (2011) propõe um processo de teste em três estágios:

1. Testes de desenvolvimento.
2. Testes de sistema.
3. Teste de aceitação.



Qual a diferença entre processos de software e modelo de processos de software?

Qual a diferença de processos de software e modelo de processos de software?

Um processo de software é composto por um conjunto de ETAPAS que são necessárias para que ele seja produzido. Já os modelos de processos incluem as ATIVIDADES que fazem parte do processo de software.



METODOLOGIAS ÁGEIS

As metodologias ágeis, segundo Sbrocco e Macedo (2012), têm sido muito utilizadas como uma alternativa para as abordagens tradicionais. Isso acontece, pelo fato de terem menos regras, menos burocracias e serem mais flexíveis, por permitirem ajustes durante o desenvolvimento do software.

EXTREME PROGRAMMING (XP)

A metodologia XP tem, como meta, atender as necessidades dos clientes com mais qualidade, mais rapidez e de forma simples.

De acordo com Martins (2007, p. 144), “a XP dá preferência ao desenvolvimento orientado a objetos e permite trabalhar com pequenas equipes de até 12 desenvolvedores (programadores). Além do mais, essa metodologia utiliza um modelo incremental que, conforme o sistema é utilizado, novas melhorias ou mudanças podem ser implementadas.

SCRUM

Scrum é um framework para desenvolver e manter produtos complexos. Em outras palavras, é uma metodologia ágil de software que concentra as suas atenções no produto final, com um rápido desenvolvimento, e nas interações dos indivíduos.



METODOLOGIAS TRADICIONAIS X METODOLOGIAS ÁGEIS

ASPECTO	METODOLOGIA TRADICIONAL	METODOLOGIA ÁGIL
Objetivo	Orientado por atividades e foco nos processos.	Orientado por produto e foco nas pessoas.
Projeto	Estável e inflexível a mudança	Adaptável a mudanças.
Tamanho	Qualquer tamanho, efetivo em projetos de longa duração.	Pequeno, mas pode ser usado em projetos de maior porte.
Gerente de projeto	Controle total do projeto.	Papel de facilitador ou coordenador.
Equipe do Projeto	Atuação com papéis claros e bem definidos em todas as atividades.	Atuação colaborativa em todas as atividades.
Cliente	Participa nas fases iniciais do levantamento de requisitos.	Parte integrante da equipe do projeto.
Planejamento	Detalhado e os envolvidos não participam.	Curto e com a participação de todos os envolvidos.

Fonte: autores

REQUISITOS

O que é um requisito?

- É uma definição detalhada e formal de uma função do sistema.
- Segundo Sommerville (2011, p. 57), “os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento”.

REQUISITOS DE USUÁRIO

- Os requisitos de usuários contemplam os requisitos funcionais e não funcionais, de forma que os usuários do sistema que não tenham conhecimentos técnicos detalhados consigam entender.

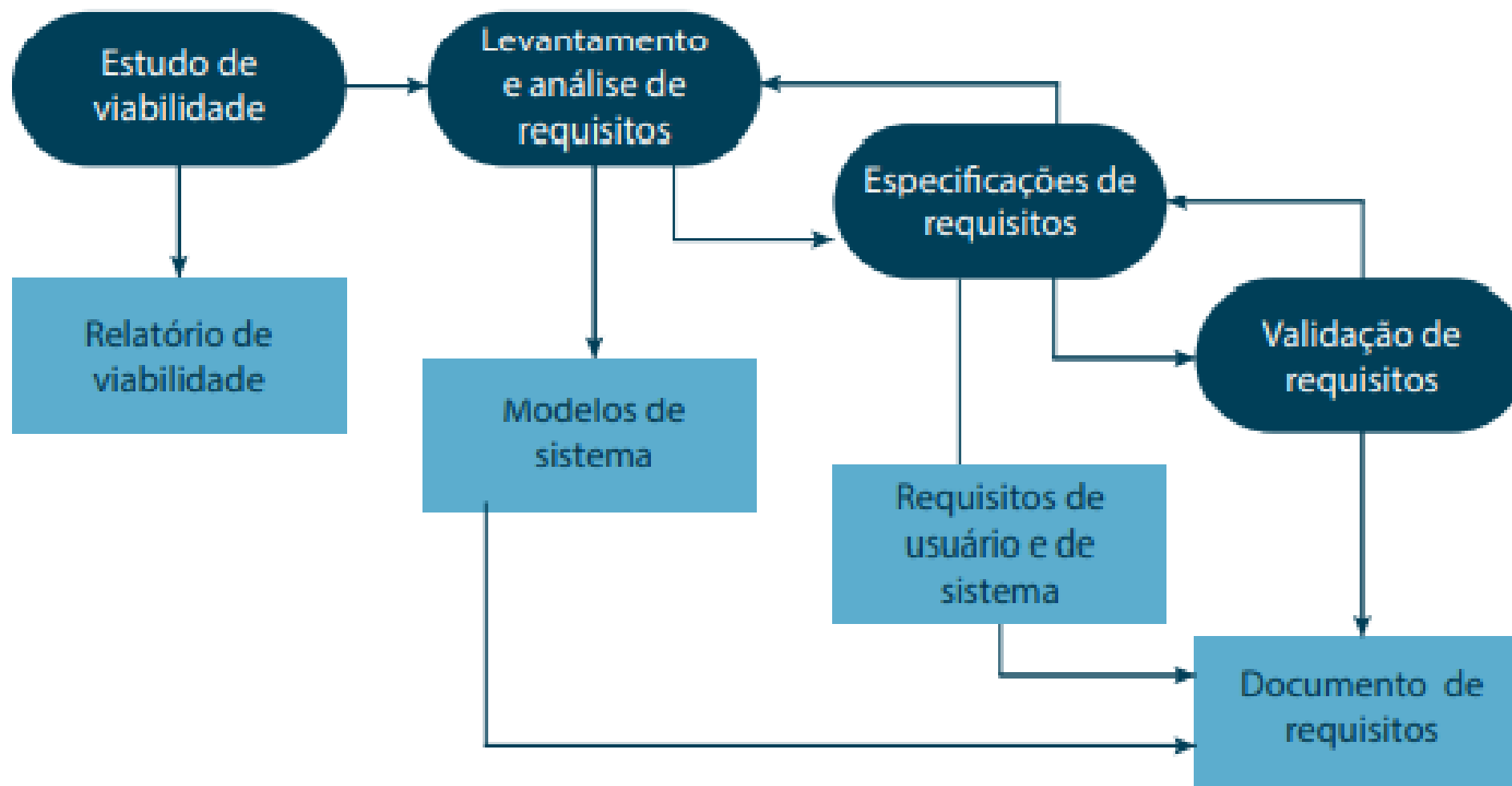
REQUISITOS DE SISTEMA

- Os requisitos de sistema são as descrições mais detalhadas dos requisitos do usuário, as quais servem como base para um contrato destinado à implementação do sistema e, portanto, devem ser uma especificação completa e consistente de todo o sistema (SOMMERVILLE, 2011).

REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

- Requisitos funcionais: Definem as funções que o sistema deve fornecer, sobre como o sistema deve reagir a entradas específicas e sobre como se comportar em determinadas situações, podendo também declarar o que o sistema não deve fazer.
- Requisitos não funcionais: São os requisitos relacionados com a utilização do software em termos de desempenho, confiabilidade, segurança, usabilidade e portabilidade, entre outros.

ENGENHARIA DE REQUISITOS



REFERÊNCIAS

IMAGEM SLIDE 13 - A AUTORA

IMAGEM SLIDE 16 - A AUTORA

IMAGEM SLIDES: 20,23, 37, 38 , 44 - LIVRO DIDÁTICO