

Nem tudo depende
do tempo, às vezes as
coisas dependem só de
uma atitude.



Antônio Dourado Cavalcanti

ENGENHARIA DE SOFTWARE

Processos de Software

Professor: Charles Leite

O processo de software

- Um conjunto estruturado de **atividades, procedimentos, artefatos e ferramentas** necessários para o desenvolvimento de um sistema de software
 - Atividades: Especificação, Projeto, Validação, Evolução

O processo de software

- Um modelo de processo de software apresenta a descrição de um processo de uma perspectiva particular, normalmente focando apenas em **algumas atividades**.

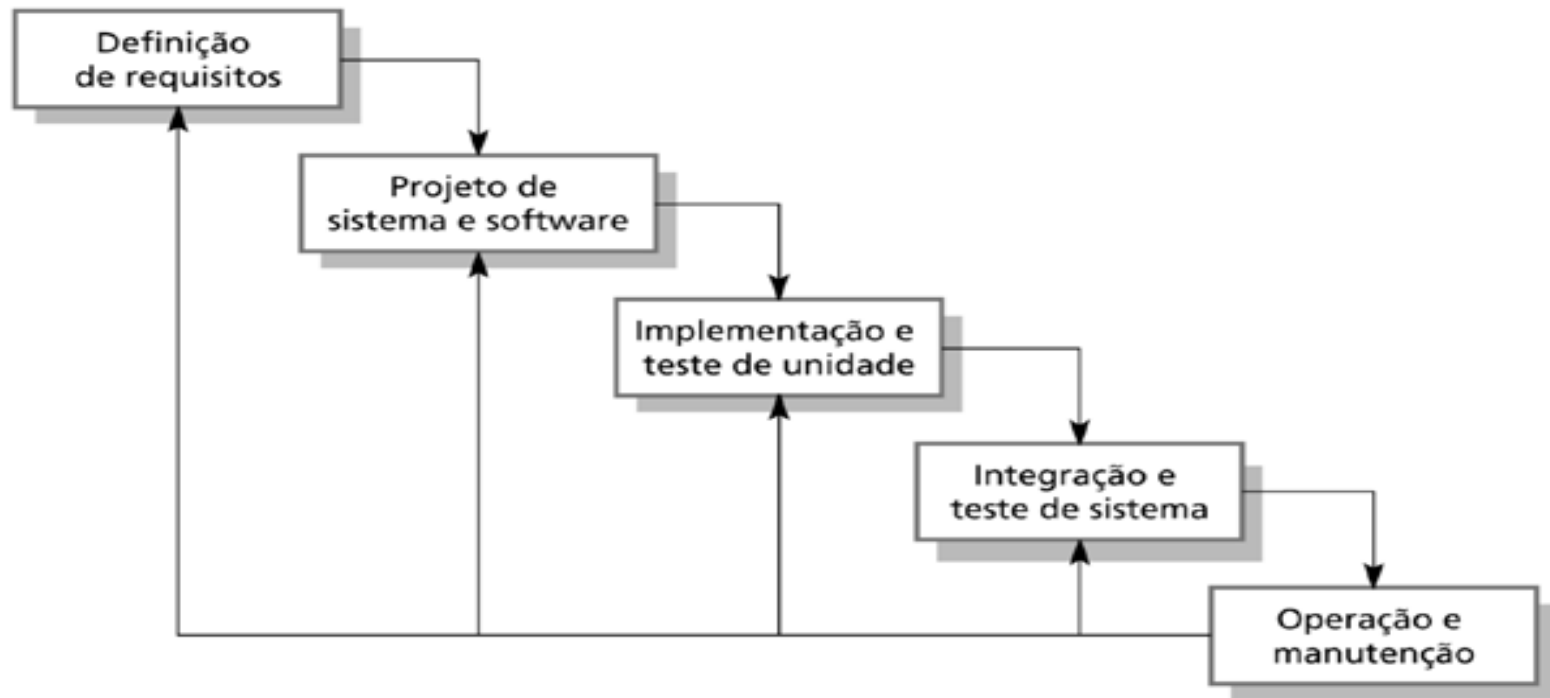
Modelos genéricos de processo de software

- O Modelo Cascata
 - Fases separadas e distintas de especificação e desenvolvimento.
- Baseada em componentes
 - O sistema é montado a partir de componentes existentes.

Modelos genéricos de processo de software

- Métodos Ágeis
 - São planejados com antecedência e o progresso é comparado
- Desenvolvimento iterativo
 - Sistema desenvolvido através de várias etapas

Modelo cascata



Ciclo de vida de software

Fases do modelo cascata



- Análise e definição de requisitos
- Projeto de sistema e software
- Implementação e teste de unidade
- Integração e teste de sistema
- Operação e manutenção

Fases do modelo cascata

- Primeiro modelo a **organizar** as atividades de desenvolvimento.
- Uma fase tem de estar completa antes de passar para a próxima.
 - Saídas das fases são acordadas contratualmente!
- Todas as fases envolvem atividades de validação

Problemas do modelo cascata

- **Particionamento inflexível** do projeto em estágios
 - Dificulta a resposta aos requisitos de mudança do cliente.
- Documentos “completamente elaborados” são necessários para fazer as transições entre estágios.
- Adequado somente quando os requisitos são bem compreendidos e quando as **mudanças** são **raras**.
 - Poucos sistemas de negócio têm requisitos estáveis.

Baseada em componentes

- Baseado em reuso sistemático onde sistemas são integrados a partir de componentes existentes.
- Estágios do processo
 - Análise de componentes;
 - Modificação de requisitos;
 - Projeto de sistema com reuso;
 - Desenvolvimento e integração.
- Esta abordagem está se tornando cada vez mais usada à medida que padrões de componentes têm surgido.
- Reuso acidental vs. Reuso planejado



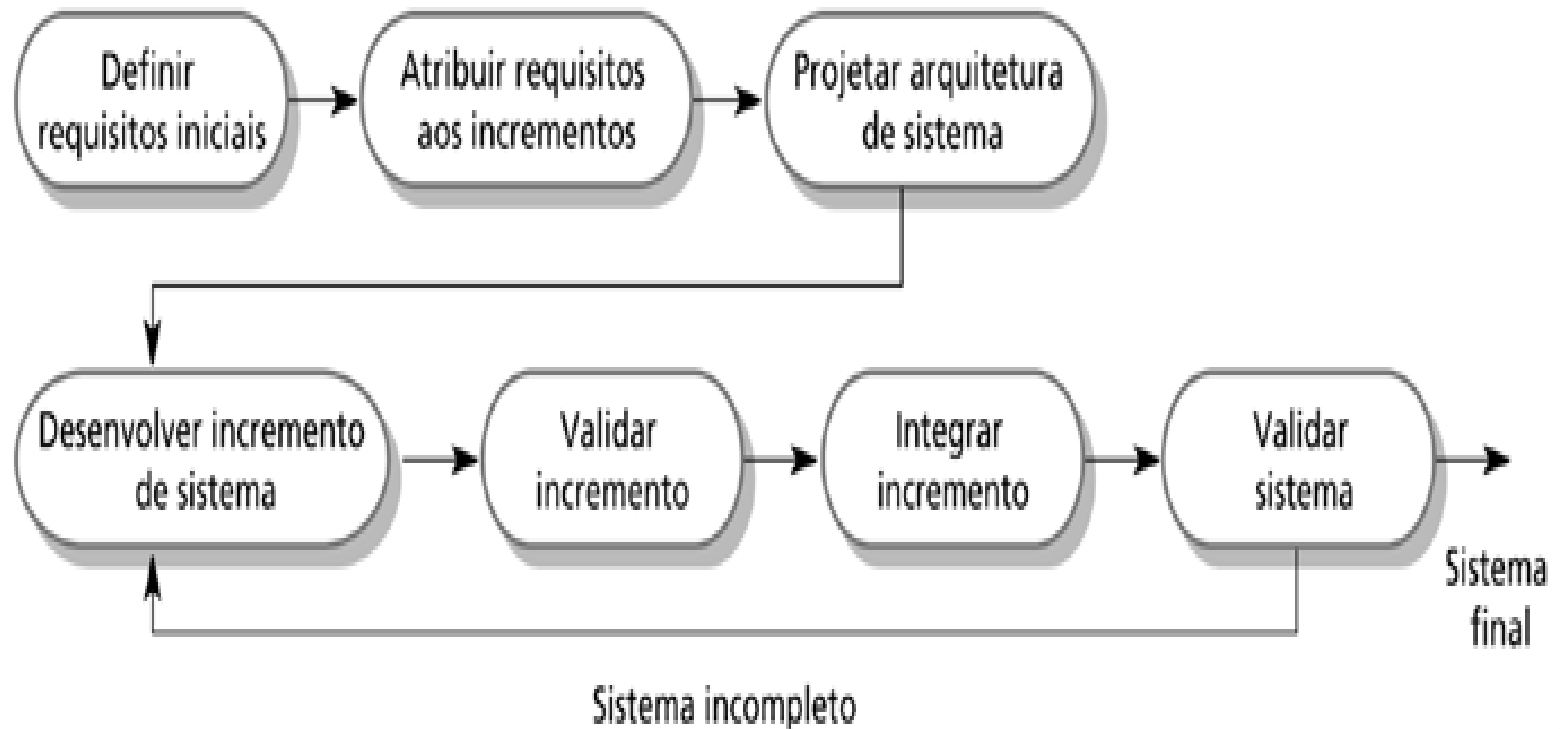
Processos Iterativos

- Requisitos de sistema **SEMPRE** evoluem no curso de um projeto
- Algum retrabalho é necessário
- A **abordagem iterativa** pode ser aplicada a qualquer um dos modelos genéricos do processo.
- Duas abordagens (relacionadas)
 - Entrega incremental;
 - Desenvolvimento espiral.

Entrega incremental

- O sistema é entregue ao cliente em incrementos
 - Cada incremento fornece parte da funcionalidade
- Os requisitos são priorizados
 - Requisitos de prioridade mais alta são incluídos nos incrementos iniciais.
- Uma vez que o desenvolvimento de um incremento é iniciado, os requisitos são congelados
 - Os requisitos para os incrementos posteriores podem continuar evoluindo (e incluir requisitos já implementados!)

Desenvolvimento incremental



Entrega incremental

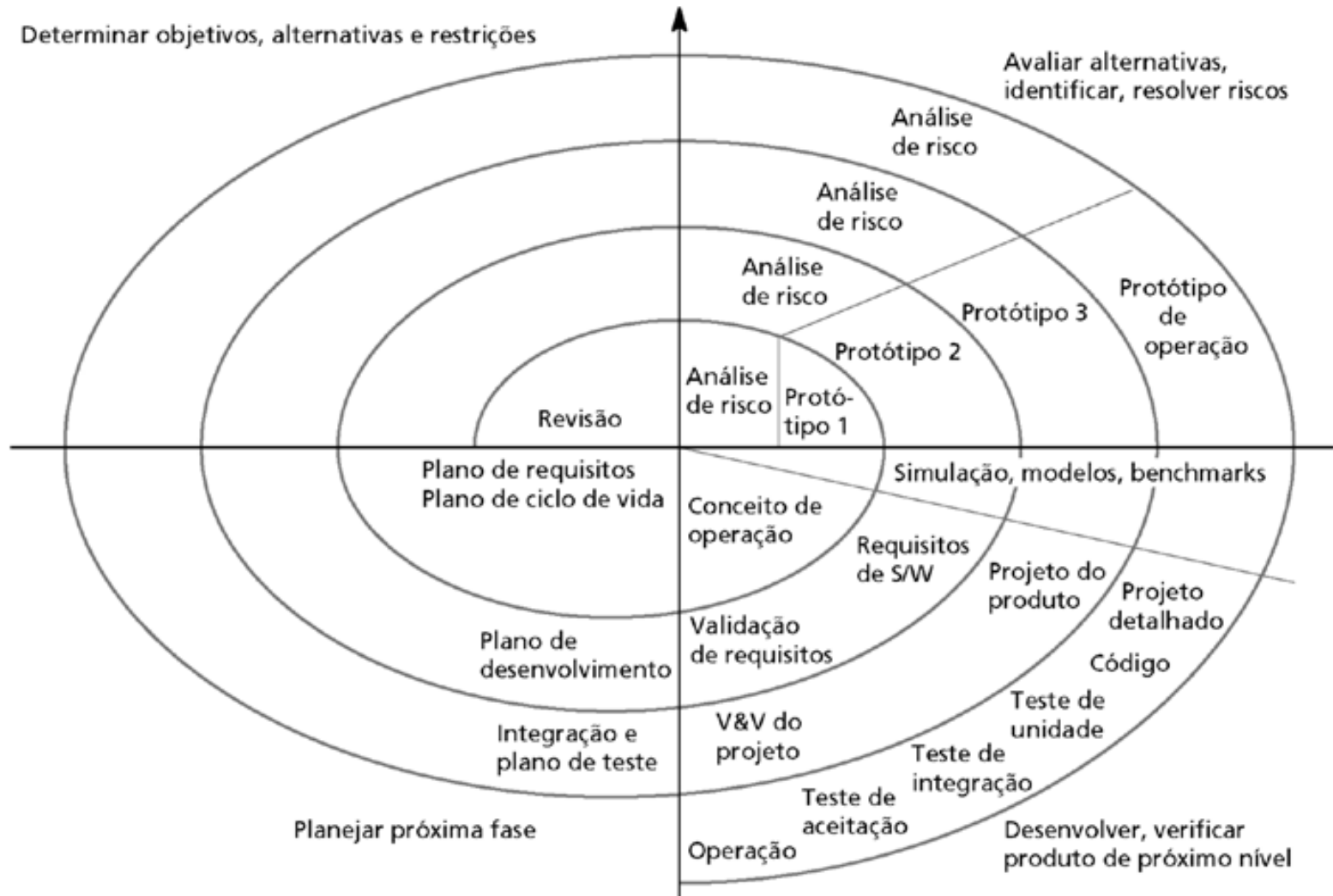
Vantagens do desenvolvimento incremental

- Incrementos podem ser entregues regularmente ao cliente e, desse modo, a funcionalidade de sistema é disponibilizada mais cedo.
- Os incrementos iniciais agem como protótipos para elicitar os requisitos para incrementos posteriores do sistema.
- Riscos menores de falha geral do projeto.
- Os serviços de sistema de mais alta prioridade tendem a receber mais testes.

Desenvolvimento espiral

- O processo é representado como uma espiral ao invés de uma sequência de atividades com realimentação.
- Cada *loop* na espiral representa uma fase no processo.
- Sem fases definidas, tais como especificação ou projeto – os *loops* na espiral são escolhidos dependendo do que é requisitado.
- Os **riscos** são explicitamente avaliados e resolvidos ao longo do processo.

Modelo espiral do processo de software Boehm



Setores do modelo espiral

- Definição de objetivos
 - Objetivos específicos para a fase são identificados.
- Avaliação e redução de riscos
 - Riscos são avaliados e atividades são realizadas para reduzir os riscos-chave.
- Desenvolvimento e validação
 - Um modelo de desenvolvimento para o sistema, que pode ser qualquer um dos modelos genéricos, é escolhido.
- Planejamento
 - O projeto é revisado e a próxima fase da espiral é planejada.

Ágil - *Extreme programming*

- Uma abordagem baseada no desenvolvimento e na entrega de incrementos de funcionalidade muito pequenos.
- Baseia-se no aprimoramento constante do código, em testes automatizados, no envolvimento do usuário na equipe e no desenvolvimento em pares.

(Rational) Unified Process

- É um (“modelo de”?) processo moderno baseado na UML
 - Tenta cobrir todos os aspectos do desenvolvimento de software
- Fortemente focado na **documentação** do sistema
- Normalmente descrito a partir de três perspectivas:
 - Uma perspectiva dinâmica que mostra as **fases** ao longo do tempo;
 - Uma perspectiva estática que mostra **atividades** de processo;
 - Uma perspectiva prática que sugere bons **princípios e práticas** de desenvolvimento

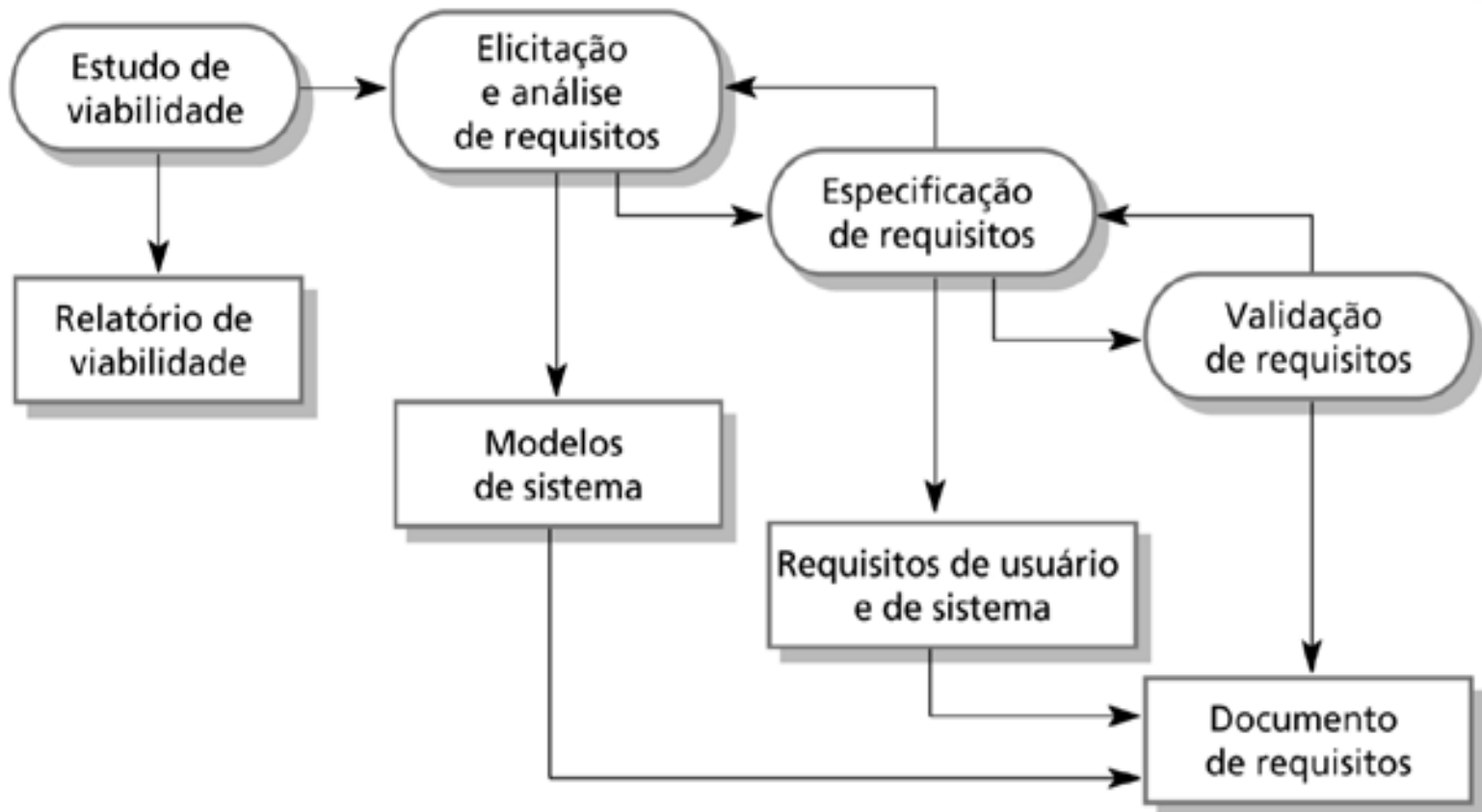
Atividades de um processo de desenvolvimento

- Especificação de software
- Projeto e implementação de software
- Validação de software
- Evolução de software

Especificação de software

- O processo para definir quais **serviços** são necessários e identificar as **restrições** de operação e de desenvolvimento do sistema.
- Processo de engenharia de requisitos
 - Estudo de viabilidade;
 - Realizado **antes** do projeto ser iniciado
 - Elicitação e análise de requisitos;
 - Especificação de requisitos;
 - Validação de requisitos.

O processo de engenharia de requisitos



- Também pode envolver a **prototipação** de partes do sistema!

Projeto e implementação de software

- É o processo de conversão da especificação em um sistema de software
- Projeto de software
 - Projetar uma estrutura de software que atenda à especificação.
- Implementação
 - Transformar essa estrutura em um programa executável.
- As atividades de projeto e implementação são fortemente relacionadas e podem ser intercaladas.

Atividades do processo de projeto

The background features a light blue and white color scheme. It is decorated with several interlocking gears of various sizes, some of which have faint text like 'time stops' and 'OR DROPS' on them. Additionally, there are blue circuit-like patterns with small circles and lines along the left and right edges of the slide.

- Projeto de arquitetura
- Especificação abstrata
- Projeto de interfaces entre componentes
- Projeto de componente
- Projeto de estrutura de dados
- Projeto de algoritmo

Programação e depuração

- É a transformação de um projeto em um programa e a remoção de defeitos desse programa.
- Programação é uma atividade pessoal – não há processo genérico de programação.
 - Há algumas práticas, porém, que são universalmente consideradas **boas**
- Programadores realizam alguns **testes** para descobrir defeitos no programa e removem esses defeitos no processo de **depuração**.

Verificação e validação de software

- Verificação e validação (V & V) têm a intenção de mostrar que um sistema está em conformidade com a sua especificação e que atende aos requisitos do cliente
- **Verificação:** “*construímos o sistema corretamente?*”
 - Exs: inspeção de código, análise estática
- **Validação:** “*construímos o sistema correto?*”
 - Exs: testes, animação de especificações
- Testes envolvem a execução do sistema com casos de teste que são derivados da especificação do sistema e de dados reais a ser processados por ele.

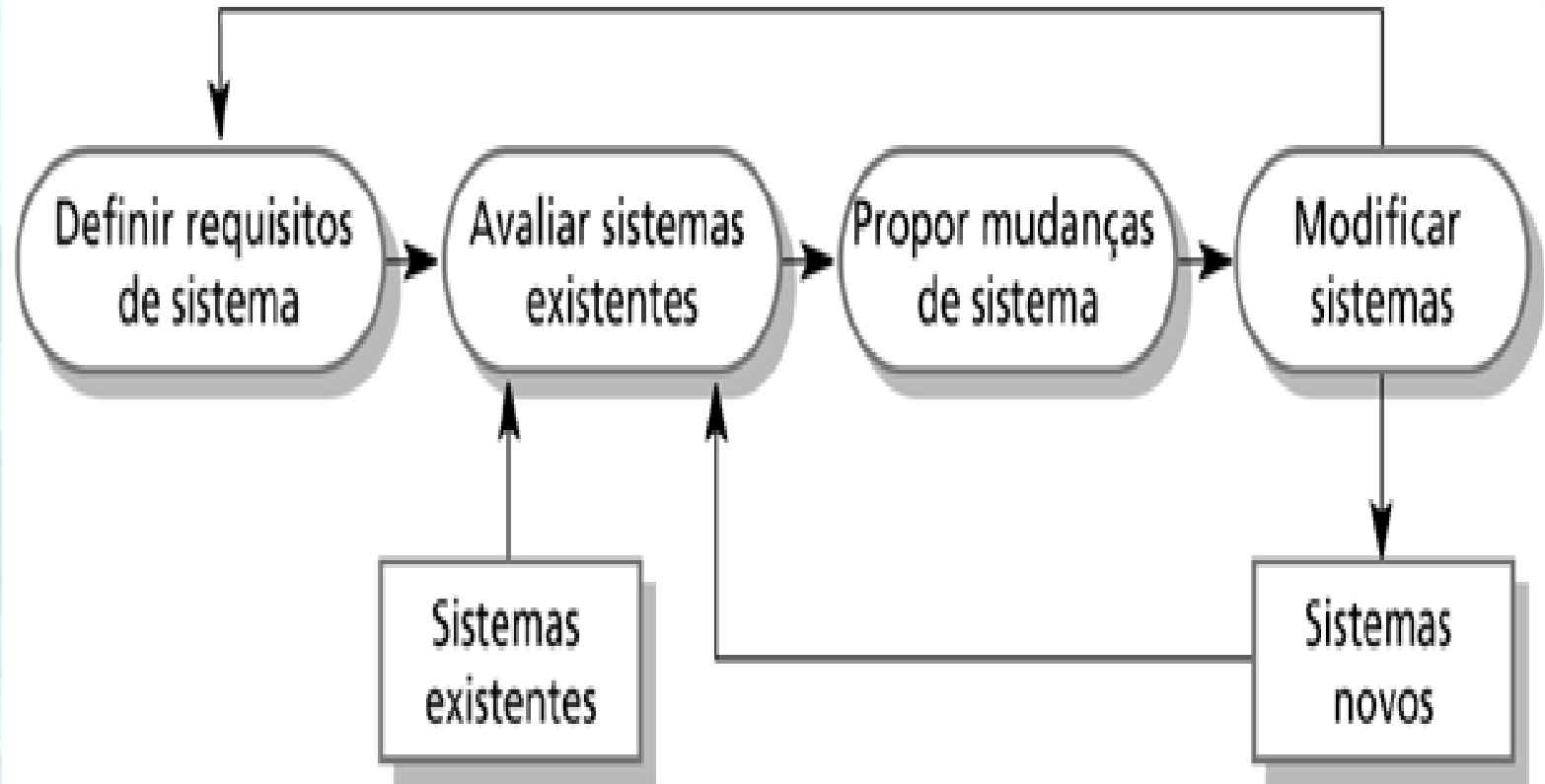
Estágios de teste

- Teste de componente ou unidade
 - Os componentes individuais são testados independentemente. Esses componentes podem ser funções ou classes de objetos, ou grupos coerentes dessas entidades.
- Teste de sistema
 - Teste de sistema como um todo. O teste das propriedades emergentes é particularmente importante.
- Teste de aceitação
 - Teste com dados do cliente para verificar se o sistema atende às suas necessidades.

Evolução de software

- O software é inerentemente flexível e pode mudar.
- Requisitos mudam devido a diversos fatores e o software deve acompanhar essas mudanças.
- Processos antigos separavam explicitamente desenvolvimento de evolução:
 - Processos e métodos iterativos (XP, RUP, Espiral) normalmente não fazem uma separação explícita.
- Evolução pode se dever a diversas razões:
 - Correções (patches)
 - Mudanças de requisitos
 - Melhoria de funcionalidades pré-existentes

Evolução de software



DÚVIDAS ...

