ARQUITETURA DE SOFTWARE

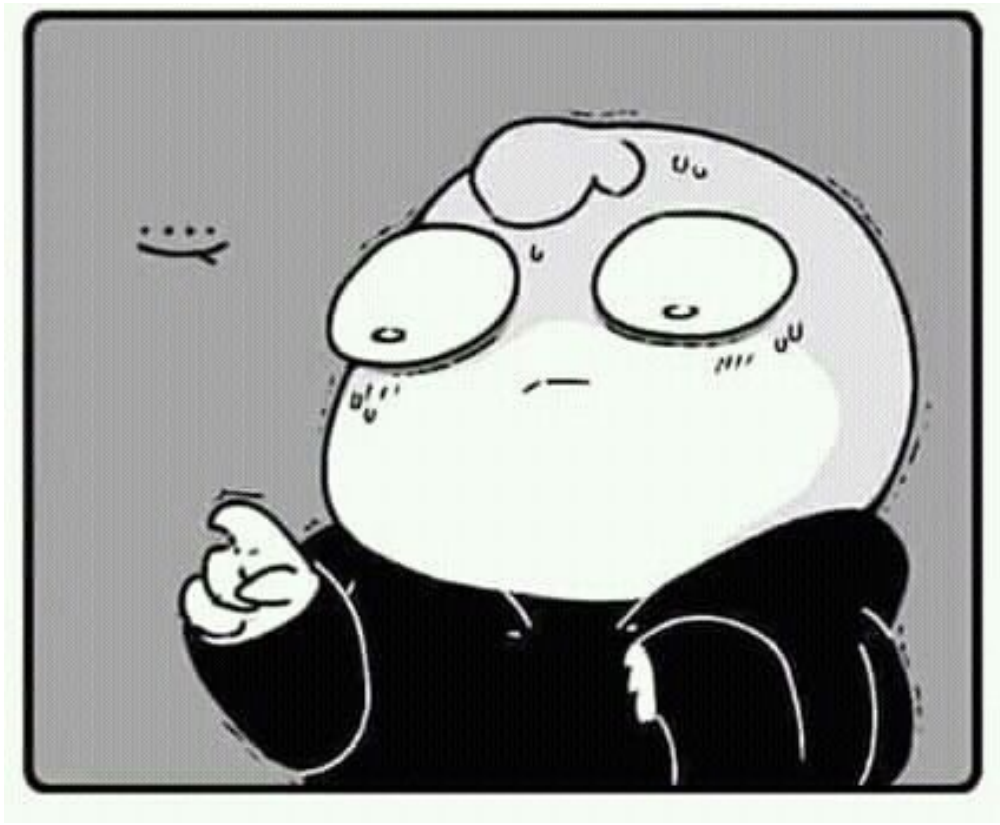
João Choma Neto

joao.choma@unicesumar.edu.br

https://github.com/JoaoChoma/arquitetura-software

Unicesumar – Maringá

MOMENTO DA ATIVIDADE 01



VAMOS ORGANIZAR OS GRUPOS

- NOME
- Registro acadêmico
- Nome do grupo

PASSOS DA ATIVIDADE

Definição do Produto:

1. Os participantes devem escolher um produto de software para implementar.
2. Pode ser uma aplicação web, um aplicativo móvel, um sistema de gerenciamento de tarefas, uma rede social simplificada, ou qualquer outra ideia de interesse do grupo

PASSOS DA ATIVIDADE

Identificação de Funcionalidades:

1. Os participantes devem listar as funcionalidades principais do produto
2. As funcionalidades podem incluir a criação de perfis de usuário, a visualização de conteúdo, a realização de ações específicas, como postar mensagens ou adicionar amigos, entre outras funcionalidades relevantes ao contexto do produto escolhido

PASSOS DA ATIVIDADE

Tomada de decisões:

1. Os participantes devem listar todas as escolhas feitas para definição do projeto com base no produto e nas funcionalidades principais do produto
2. Estas decisões abrangem:
    1. Seleção de tecnologias e frameworks
    2. Definição de estruturas de dados
    3. Algoritmos
    4. Padrões de projeto
    5. Componentes
3. Todas as decisões devem estar justificadas

PASSOS DA ATIVIDADE

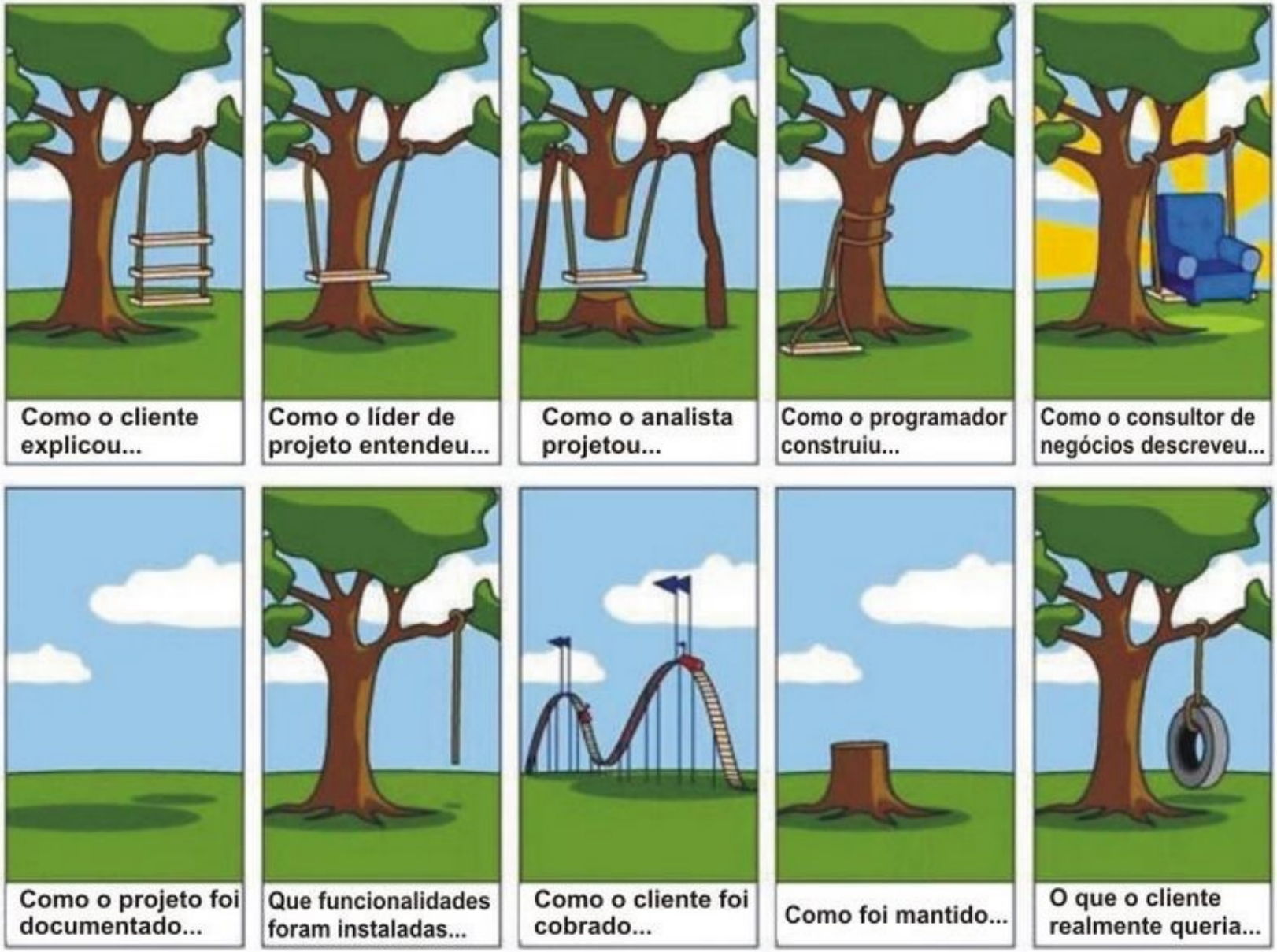
Tomada de decisões:

1. Os participantes devem listar todas as escolhas feitas para definição do projeto com base no produto e nas funcionalidades principais do produto
2. Estas decisões abrangem:
    1. Seleção de tecnologias e frameworks
    2. Definição de estruturas de dados
    3. Algoritmos
    4. Padrões de projeto
    5. Componentes
3. Todas as decisões devem estar justificadas
4. Eu sou responsável pela atividade e optei pelo padrão arquitetural MVC, por escolha própria

PASSOS DA ATIVIDADE

Organização da Arquitetura MVC:

1. Com base nas funcionalidades identificadas, os participantes devem organizar a estrutura do código seguindo o padrão MVC
2. Para esta atividade vocês devem definir quais arquivos serão criados e como irão organizar a disposição desses arquivos
    1. Se você não estiver no pc agora pode fazer no papel e tirar uma foto


Como o cliente explicou...
Como o líder de projeto entendeu...
Como o analista projetou...
Como o programador construiu...
Como o consultor de negócios descreveu...
Como o projeto foi documentado...
Que funcionalidades foram instaladas...
Como o cliente foi cobrado...
Como foi mantido...
O que o cliente realmente queria...

MODELOS DE PROCESSO DE SOFTWARE

Análise e Projeto de Sistemas

Análise

Análise (*substantivo feminino*)

1. Separação de um todo em seus elementos ou partes componentes.
2. Estudo pormenorizado de cada parte de um todo para conhecer melhor sua natureza, suas funções, relações, causas, etc.

Análise

Desenvolver **estudos** que (geralmente) partem de **problemas complexos** e que são melhor compreendidos quando **separados em partes menores**.

A etapa de análise visa investigar o problema em questão.

Problemas mal enunciados podem até ser resolvidos, porém a solução **não corresponderá às expectativas**.

Análise e **Projeto** de Sistemas

Projeto

Projeto (*substantivo masculino*)

1. Desejo, intenção de fazer ou realizar (algo) no futuro; plano.
2. Descrição escrita e detalhada de um empreendimento a ser realizado; plano, delineamento, esquema.

Projeto

Ações a serem realizadas para atingir um objetivo (levantados na análise).

O projeto propõe **uma solução para o problema** (complexo) identificado na análise.

Análise e Projeto de **Sistemas**

Sistema

Sistema de informação pode ser definido tecnicamente como um **conjunto de componentes** relacionados que **coletam**, **processam**, **armazenam** e **distribuem informações** para apoiar a tomada de decisão e o controle em uma organização.

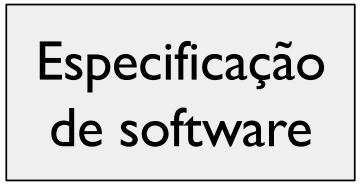
Laudon, K. C.; Laudon, J. P. **Management Information Systems**. 3. Ed. Pearson, 2014

Como desenvolver sistemas?

Como desenvolver sistemas?

Processo de Software

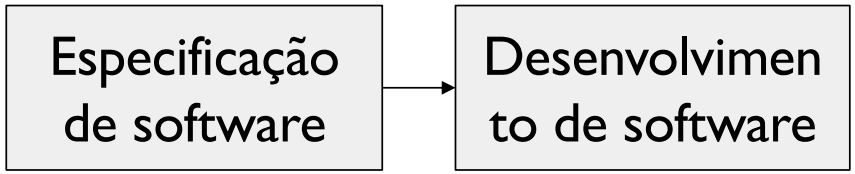
Processo de software: conjunto de atividades e resultados associados que produz um produto de software.



Especificação de software: o software a ser produzido e as restrições para a sua operação são definidos.

Processo de Software

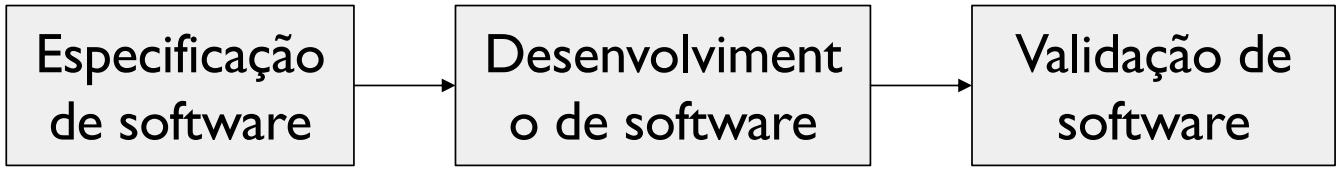
Processo de software: conjunto de atividades e resultados associados que produz um produto de software.



Desenvolvimento de software: o software é projetado e programado.

Processo de Software

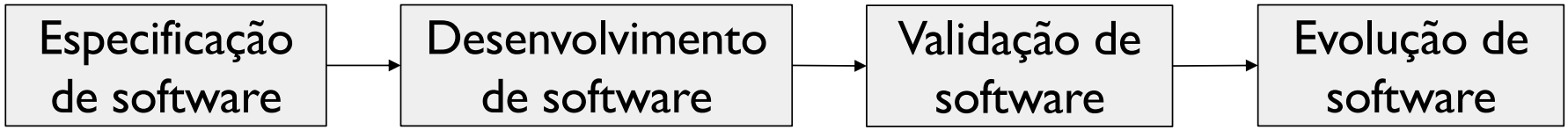
Processo de software: conjunto de atividades e resultados associados que produz um produto de software.



Validação de software: o software é verificado para garantir que é o que o cliente deseja.

Processo de Software

Processo de software: conjunto de atividades e resultados associados que produz um produto de software.



Evolução do software: o software é modificado de acordo com os novos requisitos do cliente e/ou do mercado.

Proceso de Software

Objetivos:

1. Assegurar o desenvolvimento de software com

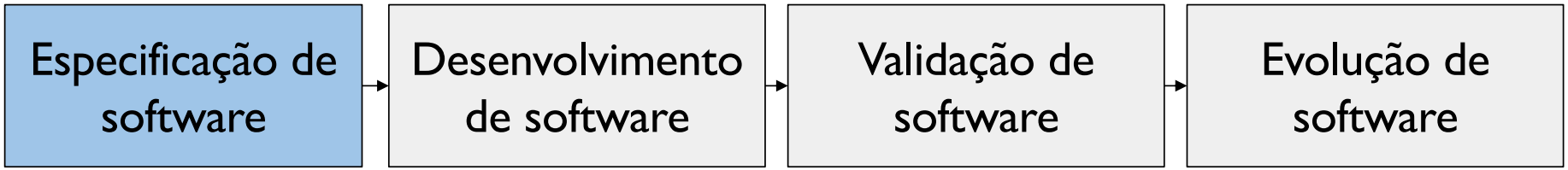
    (i) prazos e necessidade de recursos definidos,

    (ii) elevada produtividade (de forma econômica),

    (iii) qualidade assegurada.

2. Permite organizar, instrumentar, planejar, acompanhar projetos e treinar equipes.

Processo de Software



Especificação de Software

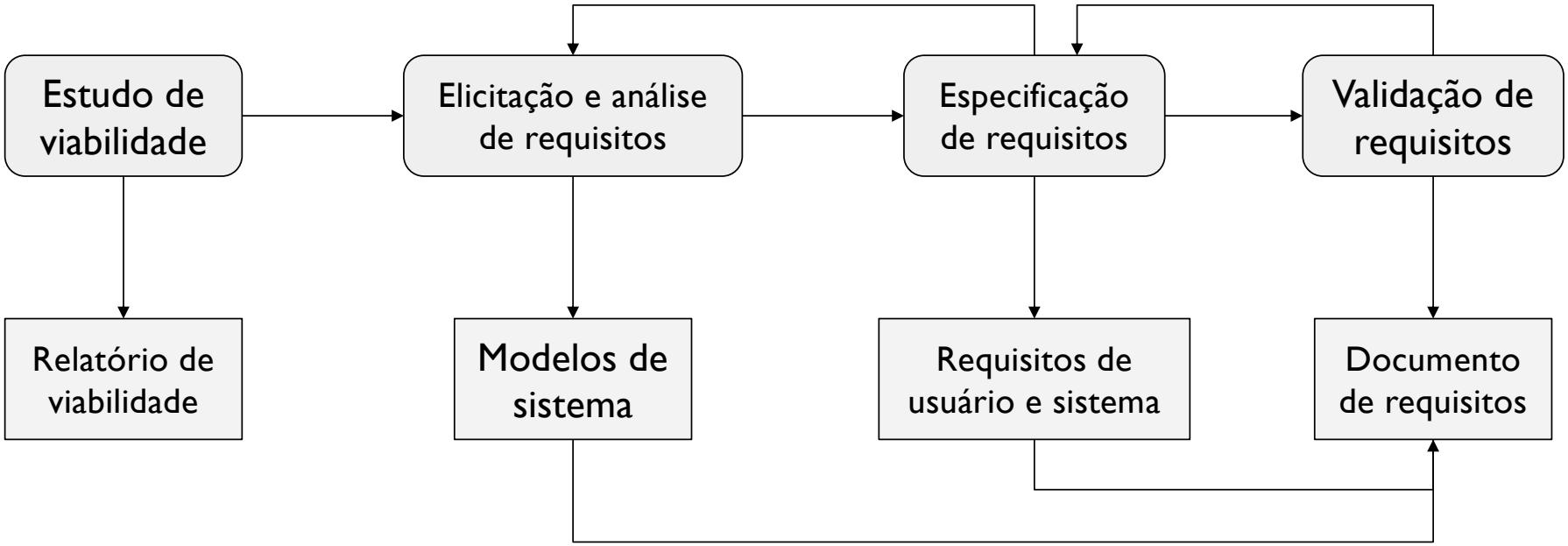
Também conhecida como Engenharia de Requisitos.

É o processo para compreender e definir quais são as **funcionalidades** necessárias e identificar as **restrições** de operação.

Etapa crítica do processo de software, pois erros nesse estágio conduzem inevitavelmente a problemas no projeto e na implementação.

O resultado é um **documento de requisitos**, que é a **especificação do sistema**.

Especificação de Software



Especificação de Software

Estudo de viabilidade

Uma avaliação é realizada para verificar se as necessidades dos usuários podem ser satisfeitas por meio das tecnologias atuais de software e hardware.

Especificação de Software

Estudo de viabilidade

A avaliação considera se o sistema terá **custo adequado** do ponto de vista comercial e poderá ser desenvolvido dentro das **restrições orçamentárias**.

O resultado é um relatório contendo informações quanto a prosseguir ou não com uma análise mais detalhada.

Especificação de Software

Elicitação de requisitos

Corresponde ao processo de **derivação de requisitos** de sistema.

Diferentes **técnicas**, como observação de sistemas existentes, discussões com usuários potenciais, análise de tarefas, podem ser utilizadas.

Desenvolvimento de um ou mais modelos de sistema ou protótipo.

Especificação de Software

Especificação de requisitos

Traduz as informações coletadas durante a atividade de análise em um documento que define um conjunto de requisitos.

Validação de requisitos

Verifica os requisitos em relação ao realismo, consistência e abrangência.

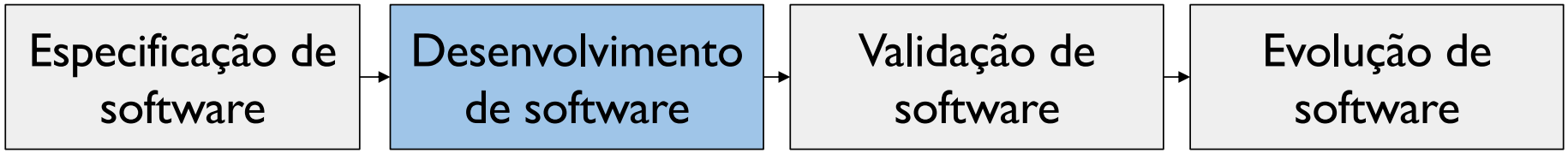
Erros no documento de requisitos são inevitavelmente descobertos.

Especificação de Software

As atividades de análise, definição e especificação podem ser intercaladas.

Em métodos ágeis, os requisitos são desenvolvidos de forma incremental, de acordo com as prioridades do usuário.

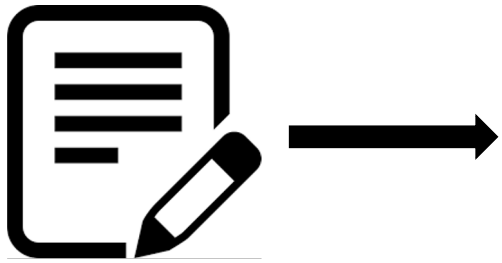
Processo de Software

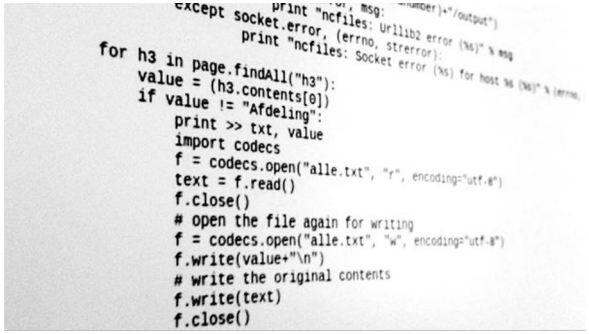


Projeto e implementação

A etapa de desenvolvimento de software corresponde ao processo de conversão de uma especificação em um sistema executável.

Envolve os processos de projeto e programação de software, além do refinamento da especificação de software (modelo evolucionário).





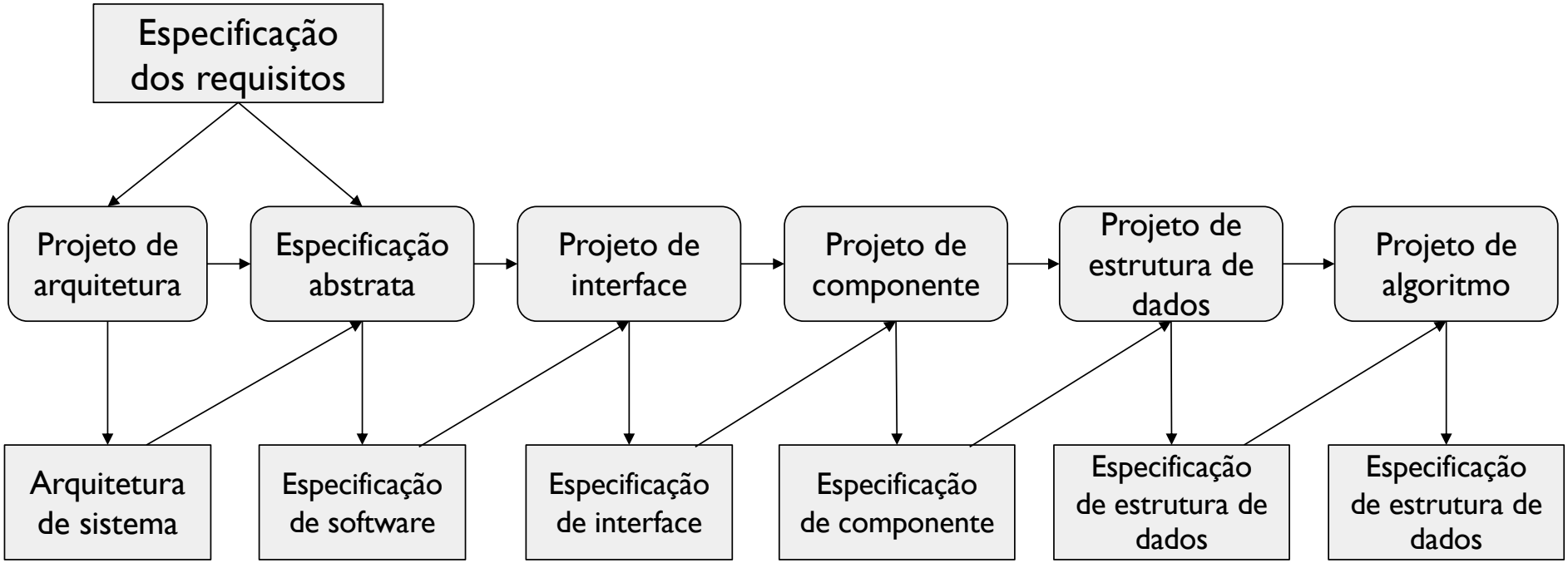
Projeto e implementação

Projeto de software é a descrição da estrutura de software a ser implementada.

- Dados do sistema,
- Interfaces entre os componentes,
- Algoritmos,
- Outros.

Desenvolvimento de vários modelos do sistema em diferentes níveis de abstração.

Projeto e Implementação



Projeto e Implementação

Projeto de arquitetura

Os subsistemas constituintes do sistema e os seus relacionamentos são identificados e documentados.

Especificação abstrata

Uma especificação abstrata dos serviços e as restrições sob as quais ele deve operar é produzida para cada subsistema.

Projeto e Implementação

Projeto de interface

Para cada subsistema, é projetada e documentada a interface com outros subsistemas.

Projeto de componente

Os serviços são alocados aos componentes e as interfaces desses componentes são projetadas.

Projeto e Implementação

Projeto de estrutura de dados

As estruturas de dados usadas na implementação do sistema são projetadas detalhadamente.

Projeto de algoritmo

Os algoritmos usados para fornecer os serviços são projetados detalhadamente.

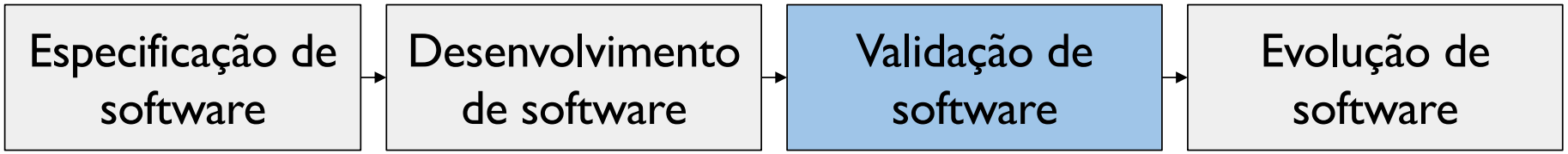
Projeto e Implementação

Com a adoção dos **métodos ágeis**, as **saídas do processo** de projeto não serão documentos de especificação separados.

As saídas **serão representadas no código do programa**.

Todas as atividades posteriores ao projeto de arquitetura serão incrementais.

Processo de Software



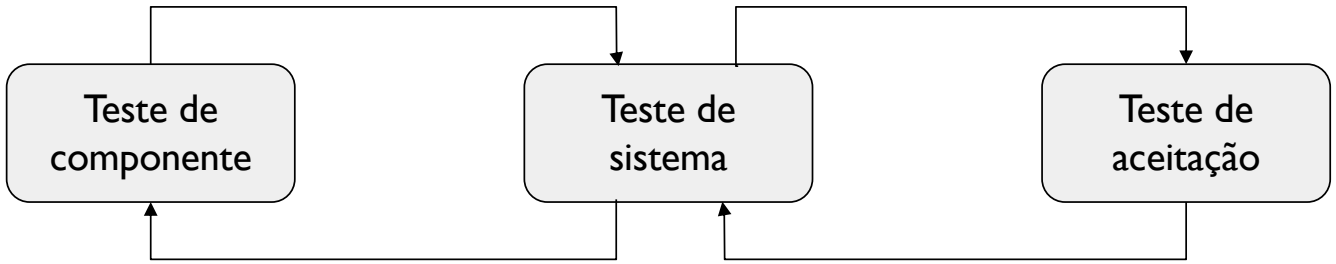
Validação de Software

Validação de software ou Verificação e Validação (V&V) destina-se a **mostrar que o sistema está em conformidade** com a sua especificação.

Verificações são **realizadas a cada estágio do processo** de software (p. ex., especificação de requisitos, projeto de sistema, código, etc.)

O maior custo de validação incorre após a implementação, quando o sistema é operacional.

Validação de Software



Validação de Software

Teste de componente (ou unidade)

Os componentes individuais são testados independentemente para garantir que operem corretamente.

- Funções,
- Classes de objetos.

Validação de Software

Teste de sistema

Os componentes são integrados para compor o sistema.

Essa etapa visa buscar **erros** que resultam **das interações não previstas** entre os componentes e **problemas de interface** de componentes.

Validação dos requisitos funcionais e não funcionais

Validação de Software

Teste de aceitação

O sistema é testado com os dados fornecidos pelo cliente do sistema, em vez de dados simulados.

O teste de aceitação pode revelar erros e omissões na definição dos requisitos do sistema.

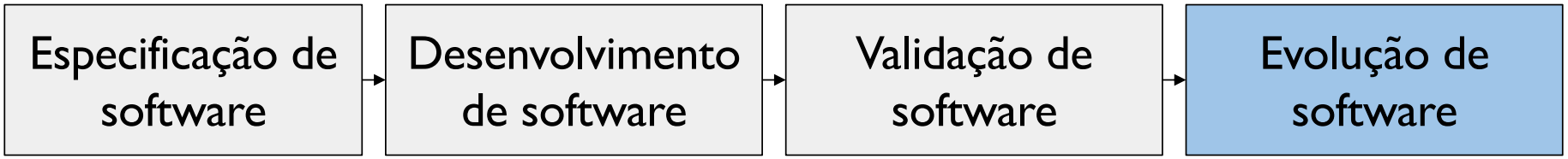
Validação de Software

O desenvolvimento e o teste de componente são intercalados.

O programador elabora seus próprios dados de teste e testa seu código.

Uma equipe independente de testadores planeja e executa os testes posteriores, baseados na especificação e projeto do sistema.

Processo de Software



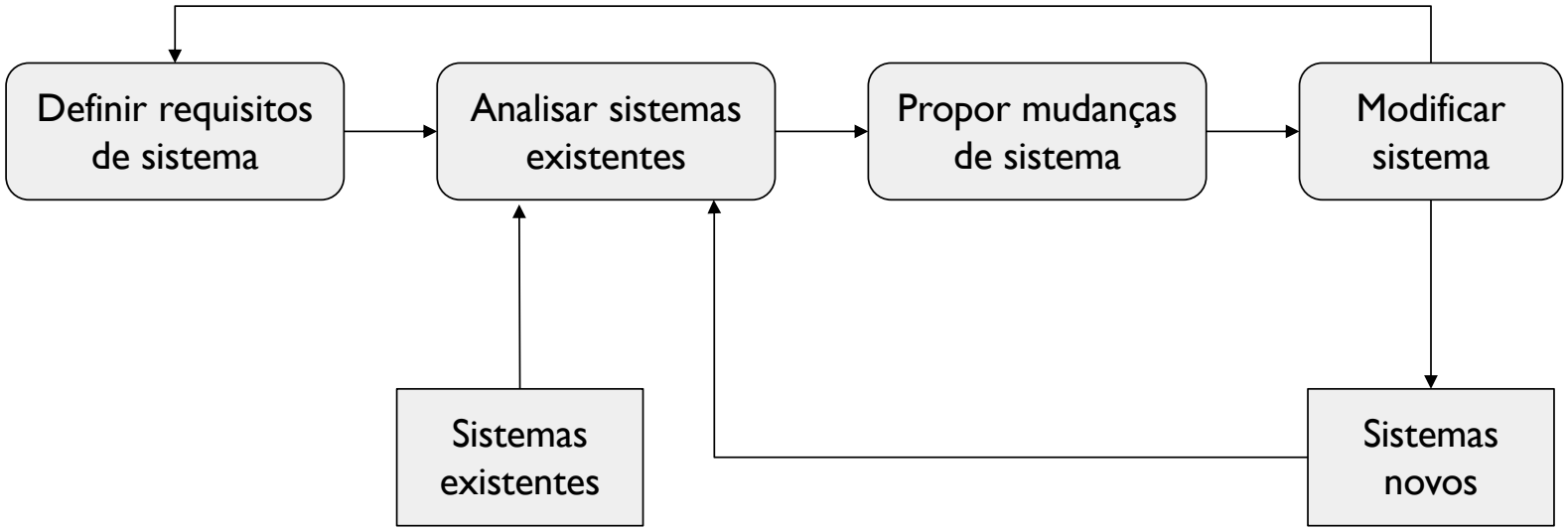
Evolução de Software

Historicamente, sempre existiu uma separação entre o processo de desenvolvimento de software e a evolução de software (manutenção).

Tal distinção tem se tornado cada vez mais irrelevante, pois, atualmente, poucos softwares são completamente novos.

Desenvolvimento e manutenção são contínuos.

Evolução do Software



Processo de Software

Diferentes tipos de sistema necessitam de diferentes processos de software.

Ex 1: Um software de aeronave deve ser completamente especificado antes do início do desenvolvimento.

Ex 2: Um software para comércio eletrônico pode ser desenvolvido juntamente com a especificação.

As atividades genéricas pode ser organizadas de diferentes maneiras.

Modelos de processo de software

É uma **descrição simplificada** do processo e representa um processo sob **determinada perspectiva**.

São utilizados para explicar diferentes abordagens do desenvolvimento de software.

Modelos genéricos:

- Modelo Cascata,
- Modelo Evolucionário,
- Engenharia de Software Baseada em Componente.

Modelo Cascata

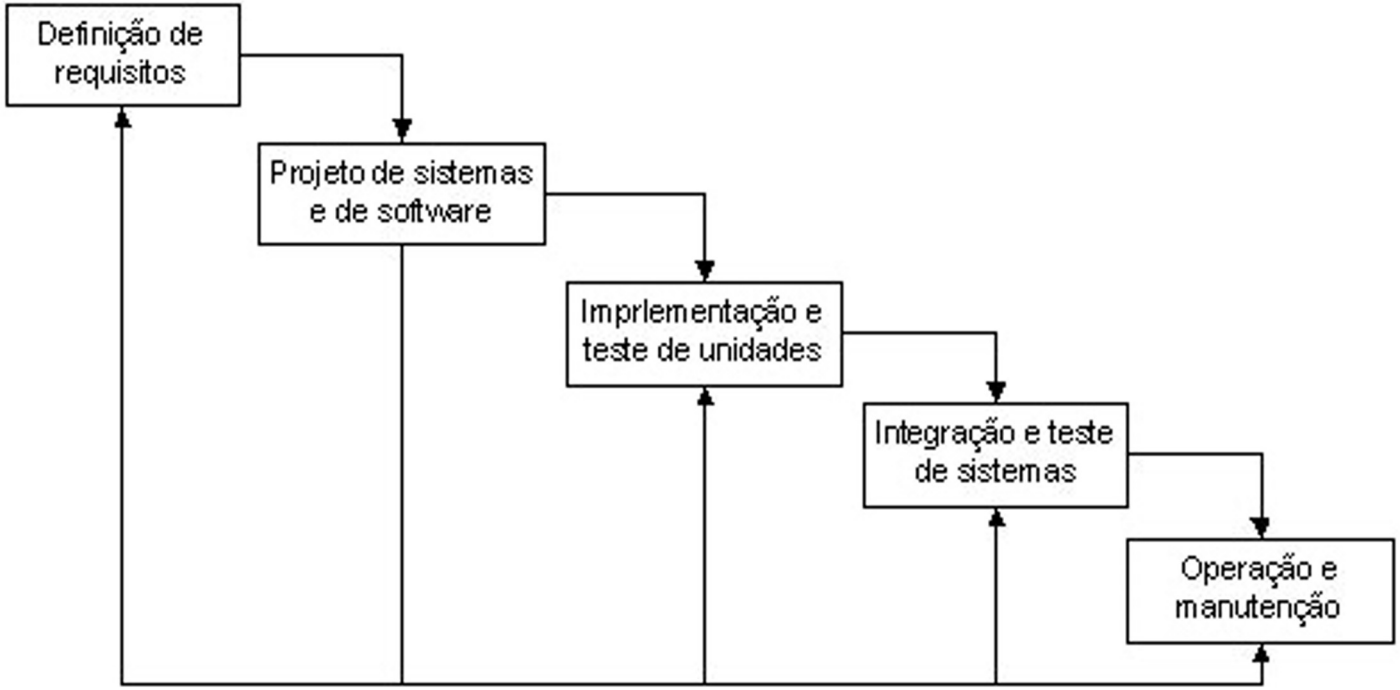
Primeiro modelo a **organizar** as atividades de desenvolvimento de software.

Atividades:

(1) Análise e definição de requisitos,
(2) Projeto de sistema,
(3) Implementação e teste de unidade,
(4) Integração e teste de sistema,
(5) Operação e manutenção.

A fase seguinte não deve começar antes que a fase anterior tenha terminado.

Modelo Cascata



Modelo Cascata

Desvantagens (problemas):

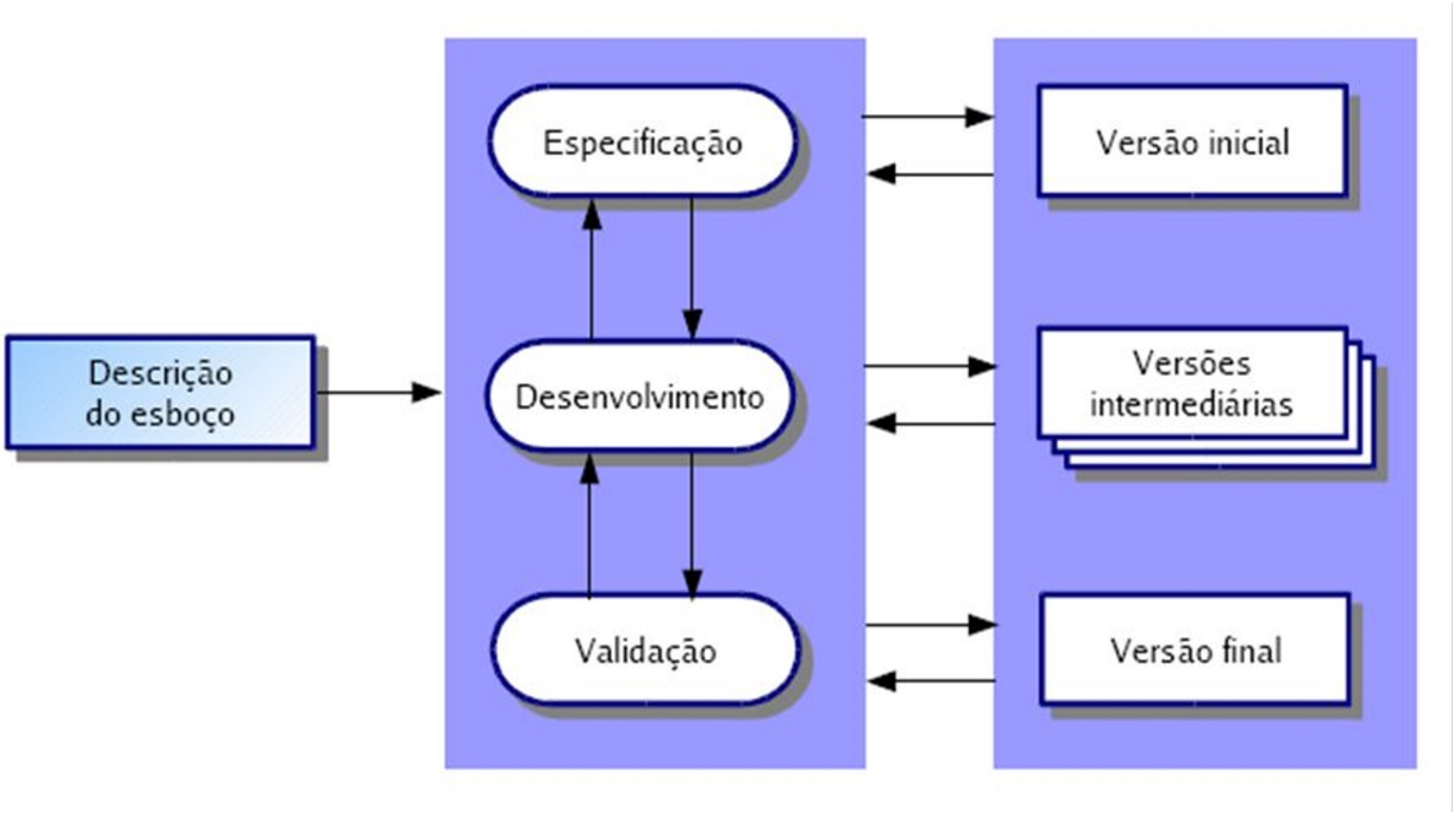
1. O resultado de cada fase envolve um ou mais documentos que são aprovados, gerando **muita documentação**.
2. A fase seguinte não deve iniciar até que a fase precedente tenha sido concluída.
3. Particionamento inflexível do projeto em estágios.
4. Apropriado somente quando os requisitos são bem compreendidos e as mudanças são raras.

Modelo Evolucionário

Desenvolvimento de uma implementação inicial, exposição do resultado aos comentários do usuário e refinamento do resultado por meio de várias versões.

As atividades de especificação, desenvolvimento e validação são intercaladas.

Modelo Evolucionário



Modelo Evolucionário

Vantagens:

- Os sistemas atendem às necessidades imediatas dos clientes.
- A especificação pode ser desenvolvida de forma incremental (usuários compreendem melhor o problema).

Modelo Evolucionário

Desvantagens:

- O progresso é medido por meio dos produtos entregues.
- A mudança contínua tende a corromper a estrutura do software.
- A incorporação de mudanças torna-se cada vez mais difícil e onerosa.

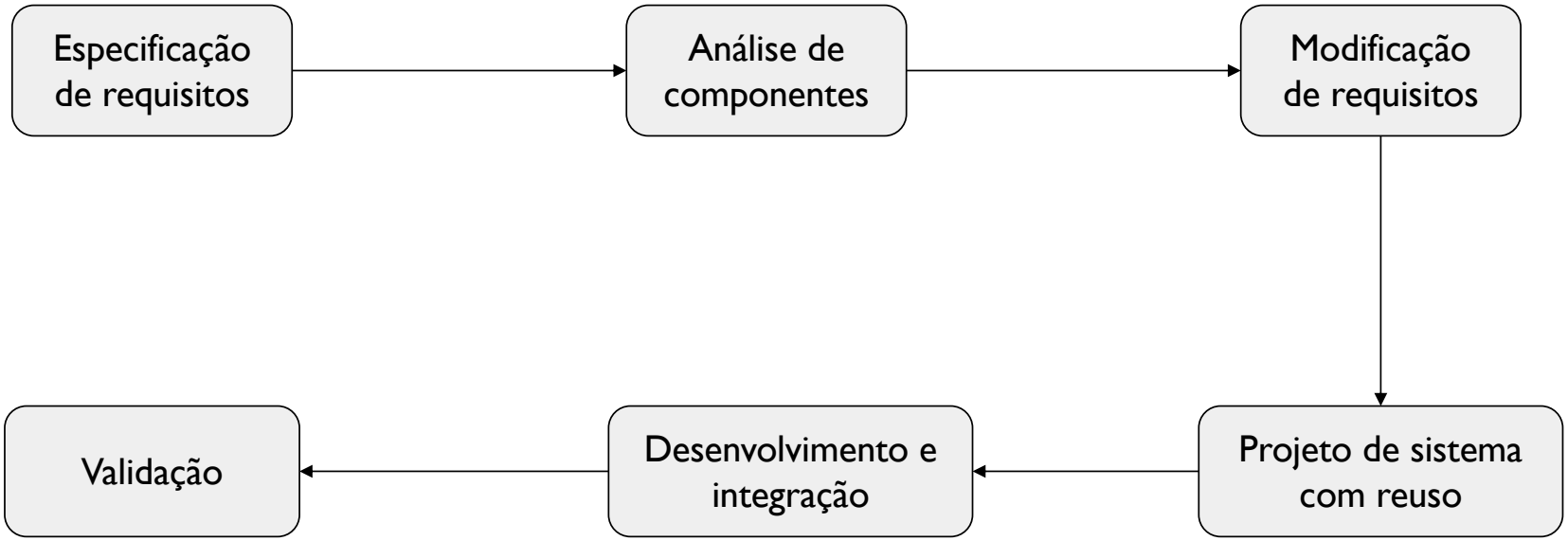
Engenharia de software baseada em componentes

Os sistemas são "desenvolvidos" a partir de componentes existentes ou sistemas COTS (Commercial-of-the-shelf).

Atividades:

(1) Análise de componentes,
(2) Modificação de requisitos,
(3) Projeto de sistema com reuso,
(4) Desenvolvimento e integração.

Engenharia de software baseada em componentes



Engenharia de software baseada em componentes

Vantagens:

- Redução da quantidade de software a ser desenvolvida, reduzindo os custos e riscos.
- Proporciona a entrega mais rápida do software.

Engenharia de software baseada em componentes

Desvantagens:

- As adequações nos requisitos são inevitáveis, podendo resultar num software que não atende às reais necessidades dos usuários.
- A evolução do software depende de novas versões dos componentes reutilizáveis, não estando sob o controle da organização que utiliza os componentes.

ARQUITETURA DE SOFTWARE

João Choma Neto

joao.choma@unicesumar.edu.br

https://github.com/JoaoChoma/arquitetura-software

Unicesumar – Maringá