



UNIVERSIDADE FEDERAL DO PIAUI – UFPI
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS - CSHNB
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO



Engenharia de Software I

Introdução a Engenharia de Software

Professora Pâmela Carvalho

29/11/22

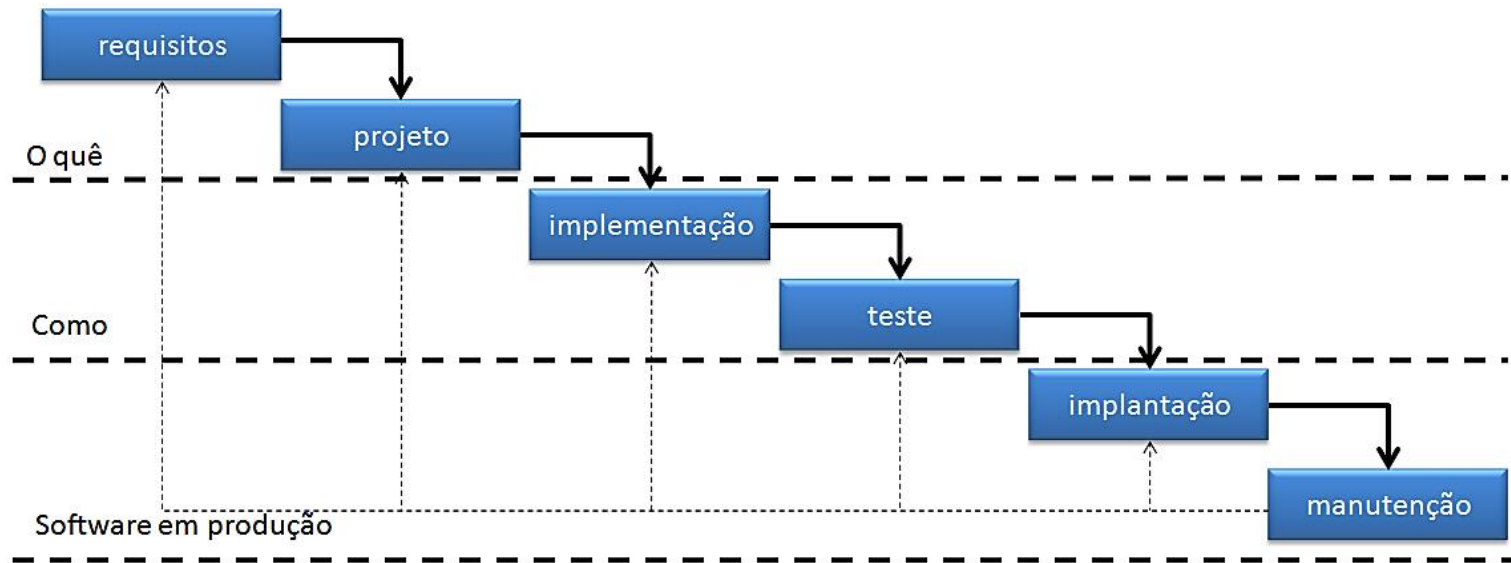
Modelo de Processo (Introdução)

- Existem vários modelos de processo de software (ou paradigmas de engenharia de software);
- Cada um representa uma tentativa de colocar ordem em uma atividade inerentemente caótica.



Modelo de Processo (Introdução)

- A **Engenharia de Software** é composta de diversos conceitos de fundamental importância na área e abrange um **processo**, um **conjunto de métodos ou práticas** e diversas **ferramentas** que possibilitam aos profissionais desenvolverem software de alta qualidade.

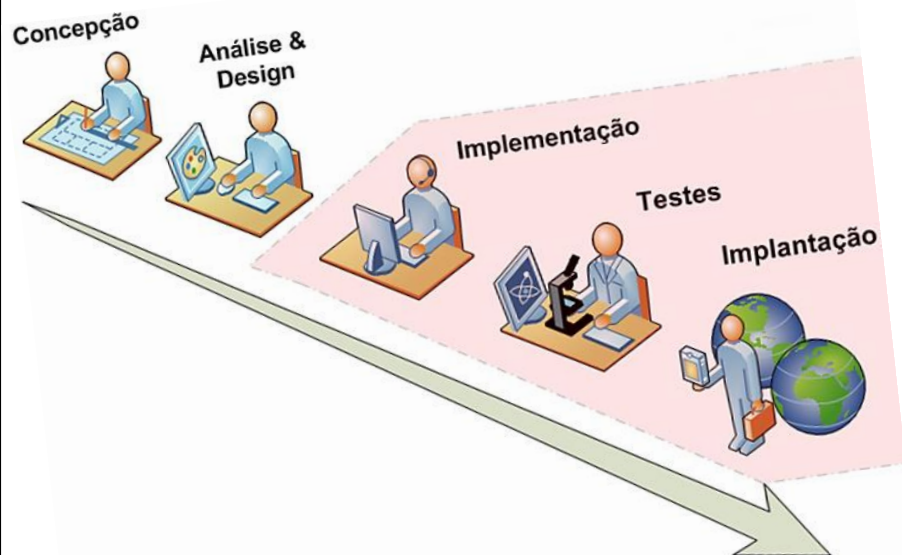


- Ex: Modelo de Processo Cascata

Modelo de Processo (Introdução)

- **O que é um processo de software?**
 - Pode ser visto como o conjunto de atividades, métodos, práticas e transformações que guiam pessoas na produção de software.

Um processo eficaz deve, claramente, considerar as relações entre as atividades, os artefatos produzidos no desenvolvimento, as ferramentas e os procedimentos necessários e a habilidade, o treinamento e a motivação do pessoal envolvido.



| Elementos que compõem um processo de software | |
|---|-------------------------|
| Processo de Software | |
| Processos | |
| Atividades | |
| | Pré-atividades |
| | Sub-atividades |
| | Artefatos |
| | Insumos |
| | Produtos |
| Recursos | |
| | Recursos Humanos |
| | Ferramentas de Software |
| | Hardware |
| Procedimentos | |
| | Métodos |
| | Técnicas |
| | Roteiros |

Modelo de Processo (Introdução)

- A criação de um software bem-sucedido se dá através de um **processo adaptável e ágil que conduzirá a um resultado** de alta qualidade, atendendo às necessidades dos clientes que de fato usarão o software.

São fatores que influenciam a definição de um processo:

- Tipo de Software (p.ex., sistema de informação, sistema de tempo real etc),
- Paradigma (estruturado, orientado a objetos etc),
- Domínio da Aplicação, Tamanho e Complexidade, Características da Equipe etc.

Modelo de Processo (Normas e Modelos de qualidade)

- Para apoiar a definição de processos, diversas **normas e modelos de qualidade de processo de software** foram propostas, dentre elas:
 - ISO 9001,
 - ISO/IEC 12207,
 - ISO/IEC 15504,
 - CMM e
 - CMMI.
- O objetivo dessas normas e modelos de qualidade é **apontar características que um bom processo de software tem de apresentar**, deixando a organização livre para estruturar essas características segundo sua própria cultura.



Modelo de Processo (Normas e Modelos de qualidade)

1. Normas Gerais da Qualidade

- **ISO 9001** – Sistema de Gestão da Qualidade

2. Normas Específicas da Melhoria de Processo de Software

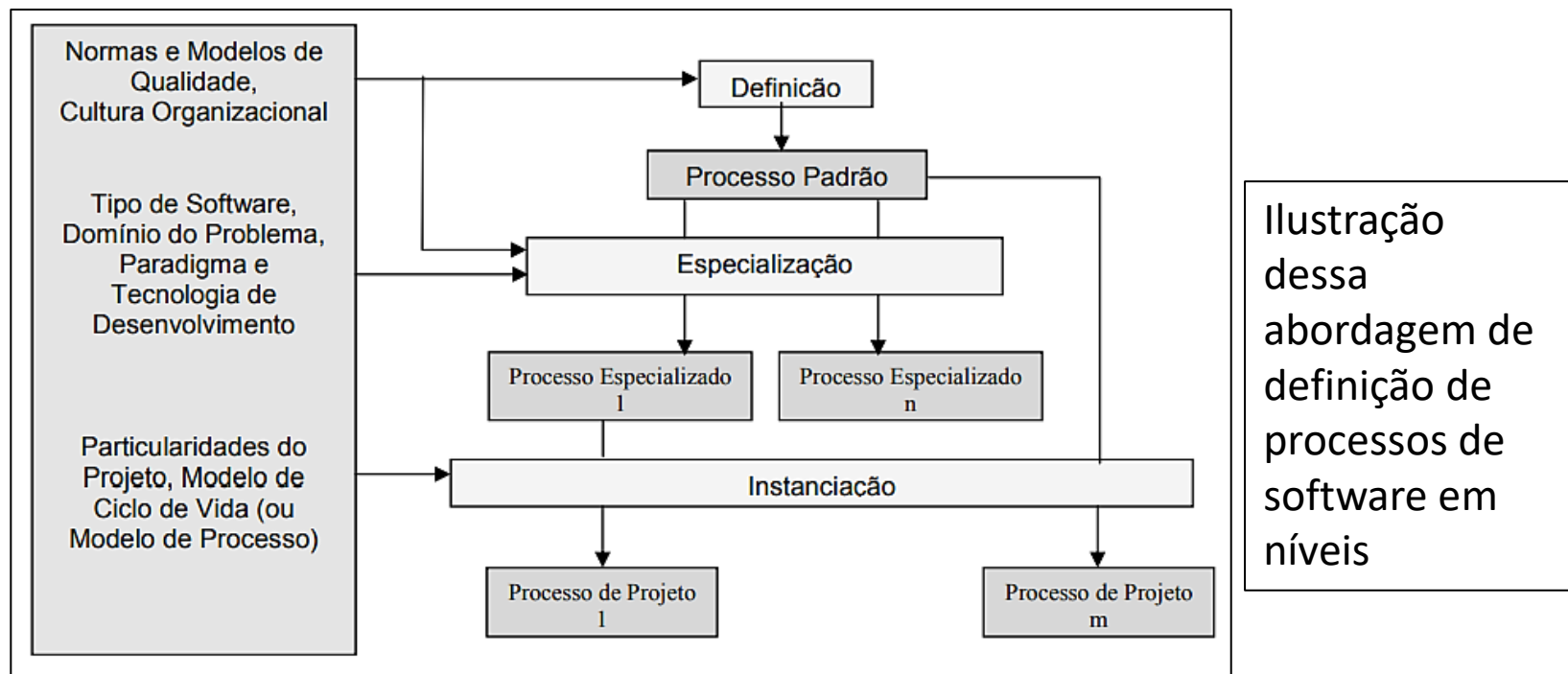
- **ISO/IEC 12207** – Engenharia de sistemas e software - Processos de ciclo de vida de software
- **ISO/IEC 15504** – Tecnologia da informação - Avaliação de processo (*SPICE - Software Process Improvement and Capability Determination*)
- **ISO/IEC 29110** – Engenharia de software — Perfis de ciclo de vida para micro-organizações (*VSEs - Very Small Entities*)

3. Modelos Específicos para Melhoria de Processo de Software

- **CMMI** – *Capability Maturity Model Integration* (The CMMI Institute, USA)
- **MoProSoft** – *Modelo de Processos para la Industria de Software* (PROSOFT, México)
- **MPS** – Modelo MPS para Software e Serviços (Softex/Programa MPS.BR, Brasil)

Modelo de Processo (Normas e Modelos de qualidade)

- Usando essas **normas e modelos de qualidade**, em uma abordagem de definição de processos em níveis, é possível definir processos para **projetos específicos**, que levem em consideração as **particularidades de cada projeto**, sem, no entanto, desconsiderar aspectos importantes para se atingir a qualidade do processo.



Modelo de Processo (Camada da ES)

- A engenharia de software está fortemente ligada com a noção de qualidade. A figura abaixo mostra as camadas da engenharia de software.



Camadas da engenharia de software

Ferramentas

Métodos

Processo

Qualidade

Apoio automatizado
ou semi-
automatizado para
processos e métodos.

Ferramentas

Métodos

Processo

Qualidade

Como fazer?

Comunicação, análise de requisitos, modelagem de projeto, construção de programas, testes, manutenção, etc.

Ferramentas

Controle gerencial de
projetos de software

Métodos

Aplicação dos
métodos

Processo

Modelos,
documentos, dados,
relatórios,
formulários, etc.

Qualidade

Ferramentas

Métodos

Processo

Qualidade

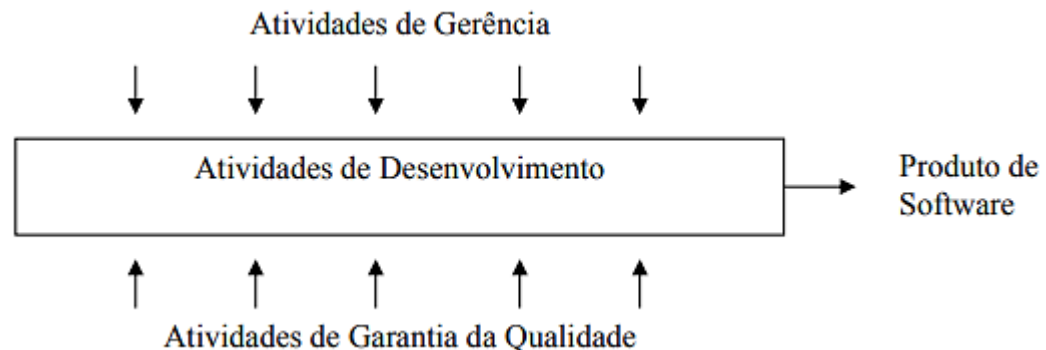
Processo contínuo de
aperfeiçoamento



Modelo de Processo (Camada da ES)



- Um **processo** não é algo rigidamente prescrito, pelo contrário, na engenharia de software o processo é uma **abordagem adaptável** que possibilita que a equipe de software selecione o trabalho a ser realizado e escolha um conjunto apropriado de ações e tarefas.



- PROCESSO:

- Ingredientes e o Modo de preparo de um RISOTO de camarão

- PRODUTO:



Modelo de Processo (Camada da ES)



- Um conceito bastante importante dentro de processo é o de **metodologia** ou **framework** que é o responsável por estabelecer uma base para um processo de engenharia de software completo através de um pequeno número **atividades estruturais aplicáveis a todos projetos de software**, independente de tamanho ou complexidade.

Modelos de Ciclos de Vida Tradicionais

Modelo Balbúrdia:

- Sistemas desenvolvidos na informalidade sem nenhum tipo de projeto ou documentação.
- O software tende a entrar num ciclo de somente duas fases:
 - Implementação;
 - implantação.

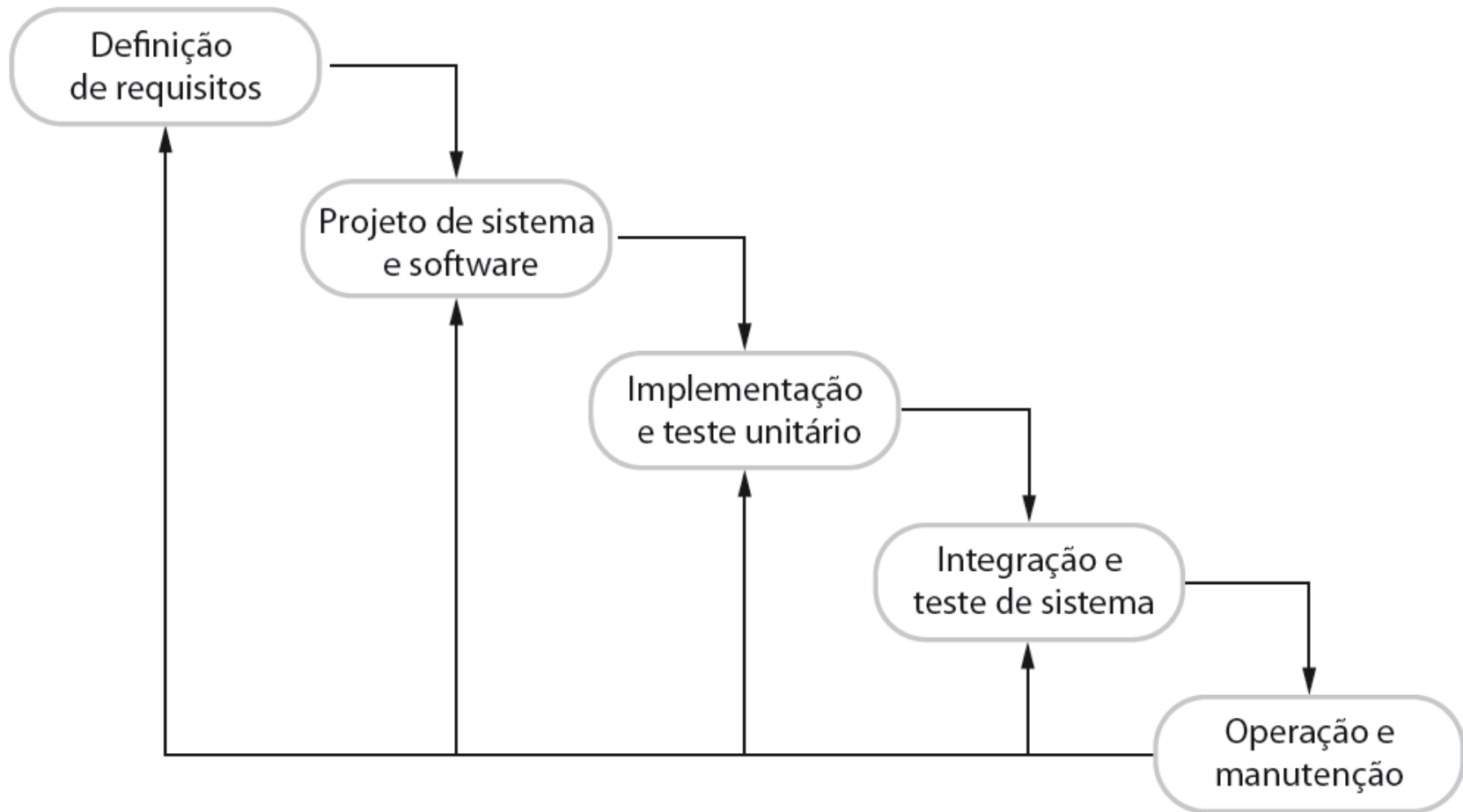


Modelos de Processo de Software

- O Modelo Sequencial Linear (também chamado Modelo Cascata)
- O Modelo de Prototipação
- O Modelo RAD (*Rapid Application Development*)
- Modelos Evolutivos de Processo de Software:
 - O Modelo Incremental
 - O Modelo Espiral
 - O Modelo de Montagem de Componentes
 - O Modelo de Desenvolvimento Concorrente
- Modelo de Métodos Formais
- Técnicas de Quarta Geração

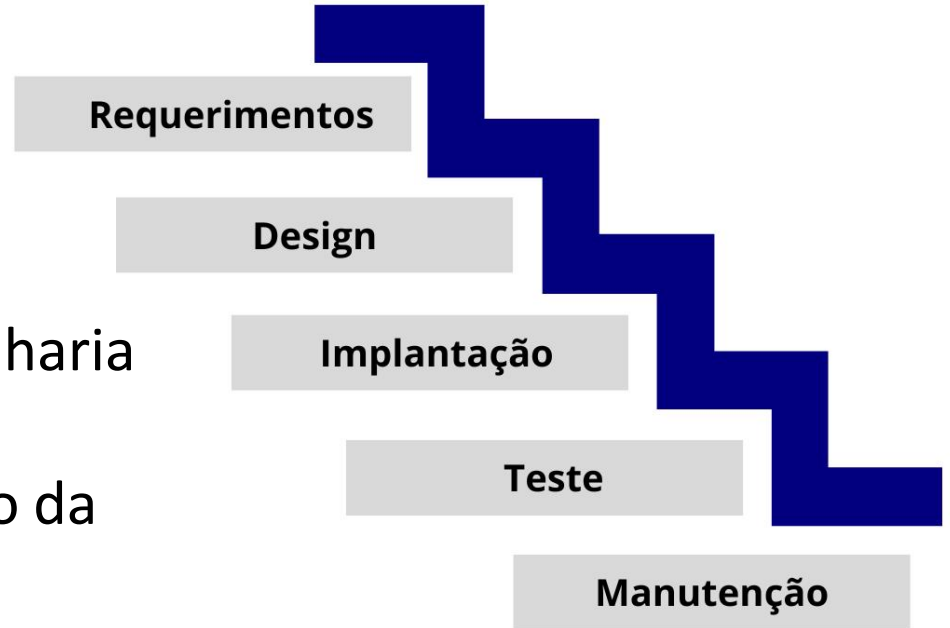
Modelo Cascata

O Modelo Cascata



Modelo Cascata

- Modelo mais antigo e o mais amplamente usado da engenharia de software;
- Modelado em função do ciclo da engenharia convencional;
- Requer uma abordagem sistemática, sequencial ao desenvolvimento de software;
- O resultado de uma fase se constitui na entrada da outra.



Fases do Modelo Cascata

- Existem fases identificadas e separadas no **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.
- O principal inconveniente do modelo cascata é a **dificuldade de acomodação de mudanças** depois que o processo já foi iniciado.
- Em princípio, uma fase precisa ser completada antes de se mover para a próxima fase.

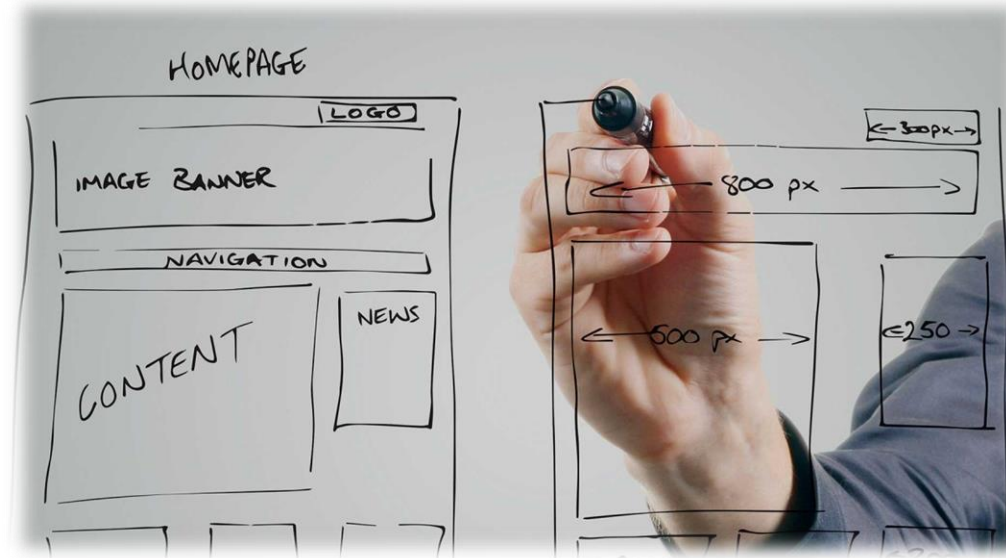
Problemas do modelo cascata

- Divisão inflexível do projeto em estágios distintos torna difícil responder às mudanças nos requisitos do cliente.
 - ✓ Por isso **esse modelo só é apropriado quando os requisitos são bem entendidos** e as mudanças durante o processo de projeto serão limitadas.
 - ✓ Poucos sistemas de negócio possuem requisitos estáveis.
- O modelo cascata é mais usado em projetos de engenharia de grandes sistemas onde o sistema é desenvolvido em vários locais.
 - ✓ Nessas circunstâncias, a natureza do modelo cascata dirigida a planos ajuda a coordenar o trabalho.

O Paradigma de Prototipação

Prototipação de software

- O objetivo é entender os requisitos do usuário e, assim, obter uma melhor definição dos requisitos do sistema.
- Possibilita que o desenvolvedor crie um modelo (protótipo) do software que deve ser construído.
- Adequado para quando o cliente não definiu detalhadamente os requisitos.



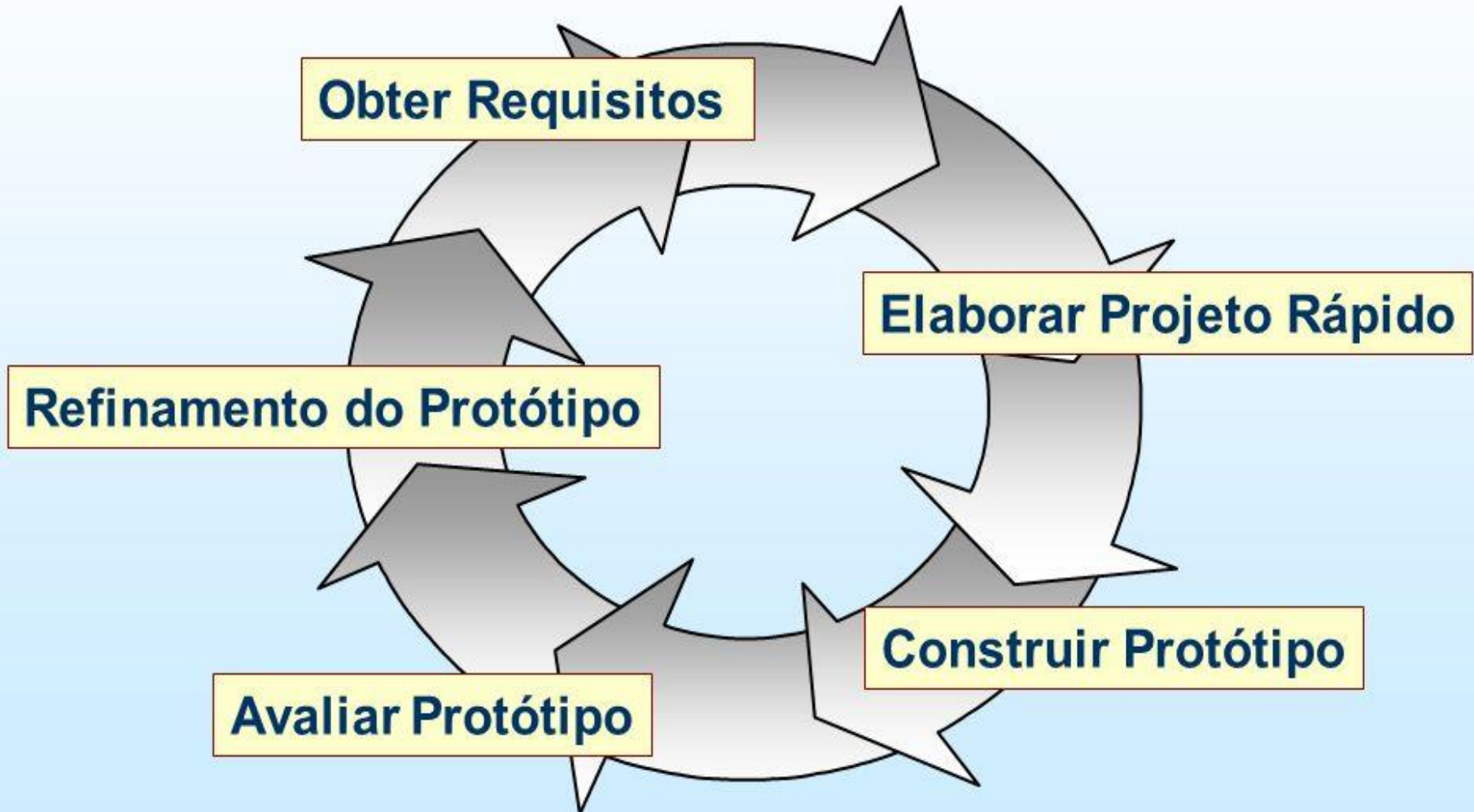
Prototipação de software

- Um protótipo é uma versão inicial de um sistema usado para demonstrar conceitos e testar opções de projeto.
- Um protótipo pode ser usado:
 - ✓ No processo de engenharia de requisitos para ajudar na elicitação e validação de requisitos;
 - ✓ Nos processos de projeto para explorar opções e desenvolver um projeto de interface de usuário;
 - ✓ No processo de testes para executar testes.

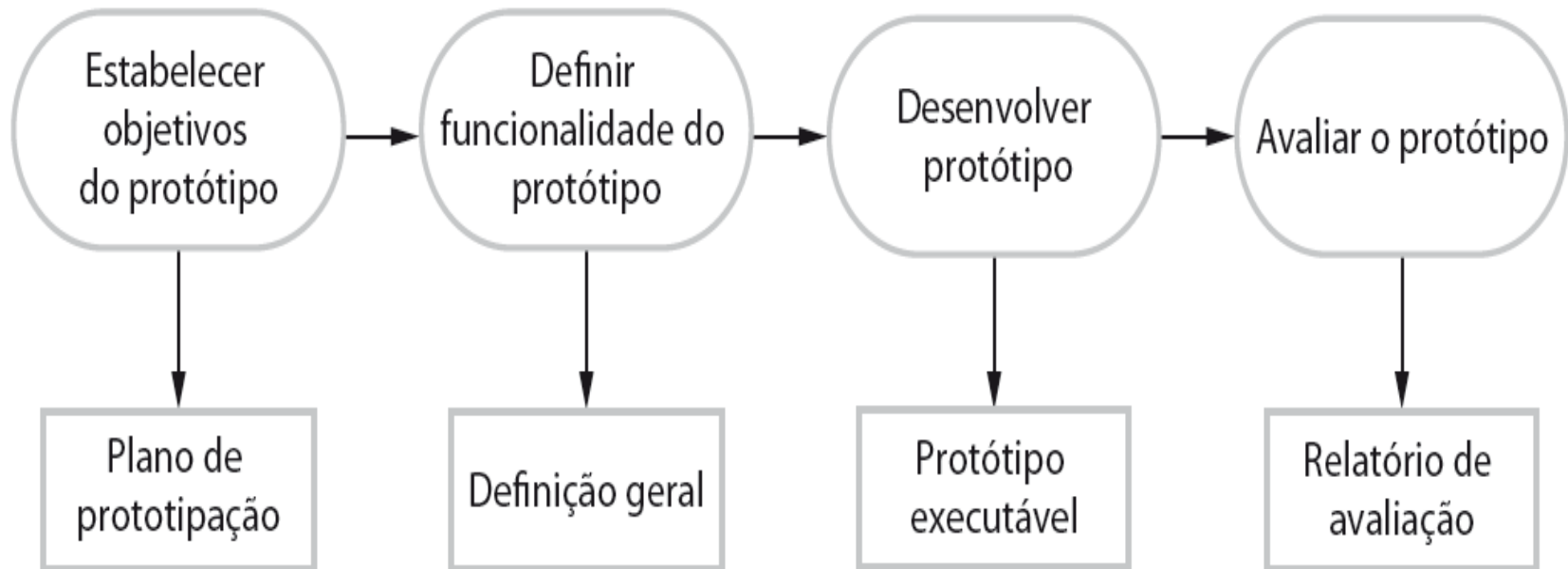
Benefícios da prototipação

- Melhoria do uso do software.
- Maior proximidade com as necessidades do usuário.
- Melhorias na qualidade do projeto.
- Maior manutenibilidade.
- Reduzir esforços de desenvolvimento.

Modelo de Prototipação



O processo de desenvolvimento de protótipo



Desenvolvimento de protótipos

- Pode ser baseado em linguagens ou ferramentas de prototipagem rápida.
- Pode deixar a funcionalidade de fora do teste.
 - ✓ A prototipação deve focar em áreas do produto que não são bem entendidas;
 - ✓ A checagem de erros e recuperação podem não estar incluídas no protótipo;
 - ✓ O foco deve ser em requisitos funcionais ao invés de não funcionais como por exemplo, a confiabilidade e a segurança.

Descarte de protótipos

- Os protótipos devem ser descartados depois do desenvolvimento, pois não são uma boa base para um sistema em produção:
 - ✓ Pode ser impossível ajustar o sistema para alcançar requisitos não funcionais;
 - ✓ Geralmente os protótipos não possuem documentação;
 - ✓ Geralmente a estrutura do protótipo é degradada por mudanças rápidas;
 - ✓ Provavelmente o protótipo não irá alcançar os padrões normais de qualidade organizacional.

Problemas com a Prototipação

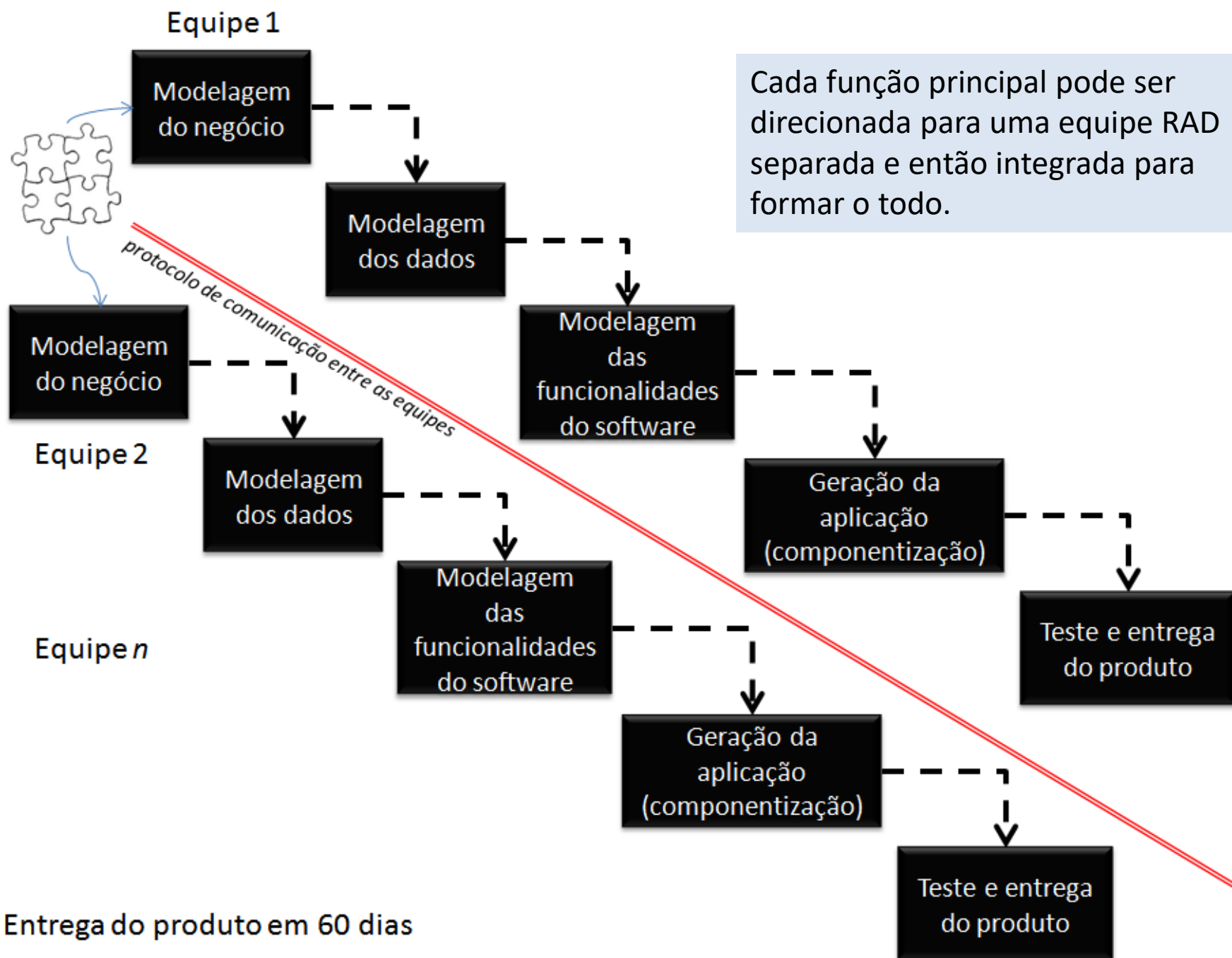
- O cliente não sabe que o software que ele vê não considerou, durante o desenvolvimento, a qualidade global e a manutenibilidade a longo prazo;
- O desenvolvedor frequentemente faz uma implementação comprometida (utilizando o que está disponível) com o objetivo de produzir rapidamente um protótipo.

Modelo RAD

Modelo RAD

- RAD (*Rapid Application Development*) é um modelo sequencial linear que enfatiza um ciclo de desenvolvimento extremamente curto;
- O desenvolvimento rápido é obtido usando uma abordagem de construção baseada em componentes.
- Os requisitos devem ser bem entendidos e o alcance do projeto restrito.
- O modelo RAD é usado principalmente para aplicações de sistema de informação.

Cada função principal pode ser direcionada para uma equipe RAD separada e então integrada para formar o todo.



Problemas com o RAD

- Exige recursos humanos suficientes para todas as equipes.
- Exige que desenvolvedores e clientes estejam comprometidos com as atividades de “fogo-rápido” a fim de terminar o projeto num prazo curto.

Modelos Evolutivos de Processo

Modelo Incremental

Modelo Espiral

Modelos Evolutivos de Processo

- Existem situações em que a engenharia de software necessita de um modelo de processo que possa acomodar um produto que evolui com o tempo.
- Modelos evolutivos são iterativos possibilitam o desenvolvimento de versões cada vez mais completas do software.

Modelo Incremental

Modelo Incremental

- O modelo incremental combina elementos do modelo cascata (aplicado repetidamente) com a filosofia iterativa da prototipação.
- O objetivo é trabalhar junto do usuário para descobrir seus requisitos, de maneira incremental, até que o produto final seja obtido.

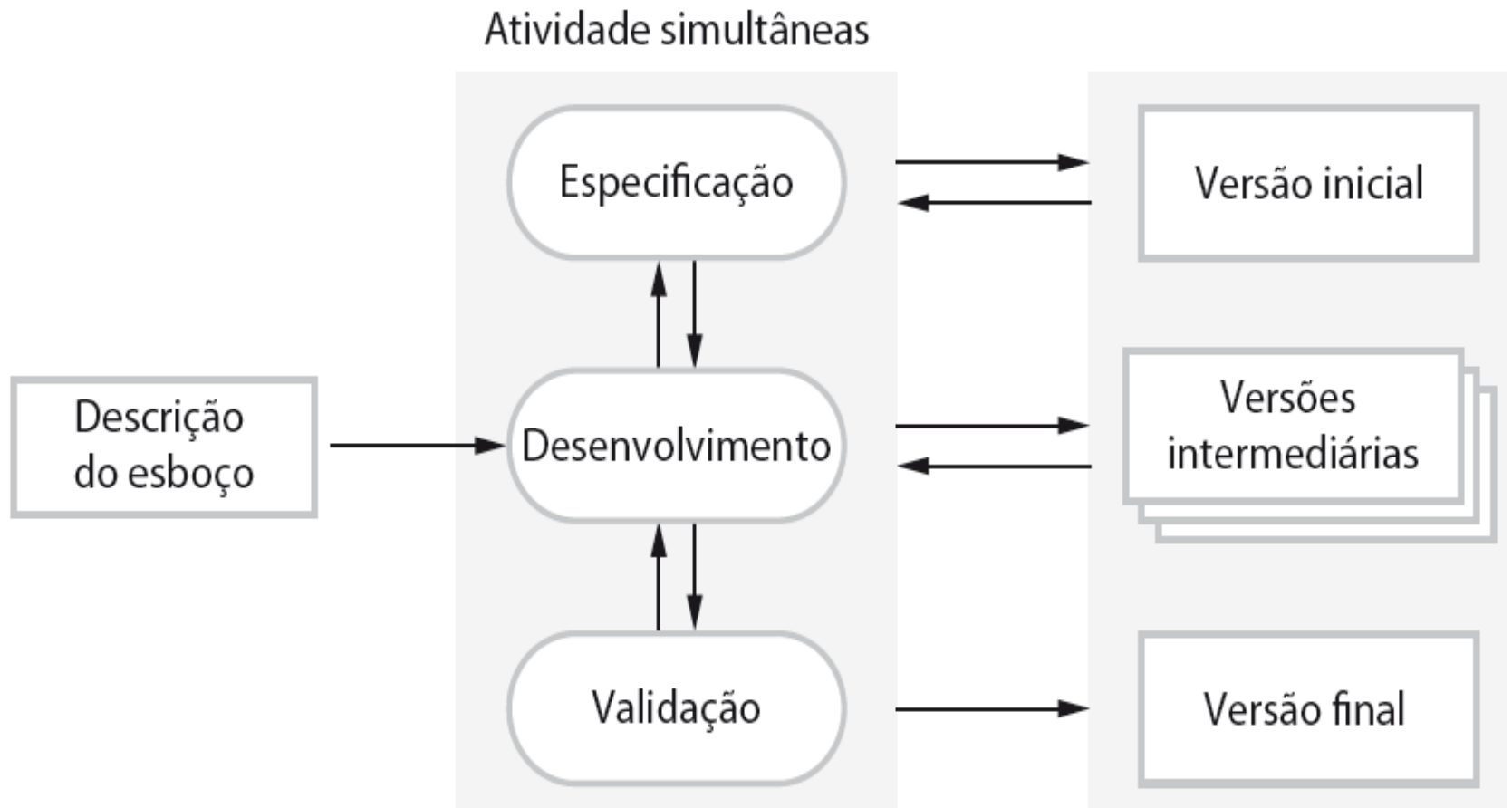
Modelo Incremental

- Ao invés de entregar o sistema em uma única entrega, o desenvolvimento e a entrega são distribuídos em incrementos, nos quais cada incremento entrega parte da funcionalidade necessária.
- Os requisitos do usuário são priorizados e os requisitos de mais alta prioridade são incluídos nos primeiros incrementos.

Desenvolvimento e entrega incremental

- Desenvolvimento incremental
 - ✓ Desenvolve o sistema em incrementos e avalia cada incremento antes de proceder com o desenvolvimento do próximo incremento;
 - ✓ Abordagem normalmente usada em métodos ágeis;
 - ✓ Avaliação feita por representantes do usuário/cliente.
- Entrega incremental
 - ✓ Implanta um incremento para uso do usuário-final;
 - ✓ Avaliação mais realística sobre o uso prático do software.

Desenvolvimento incremental



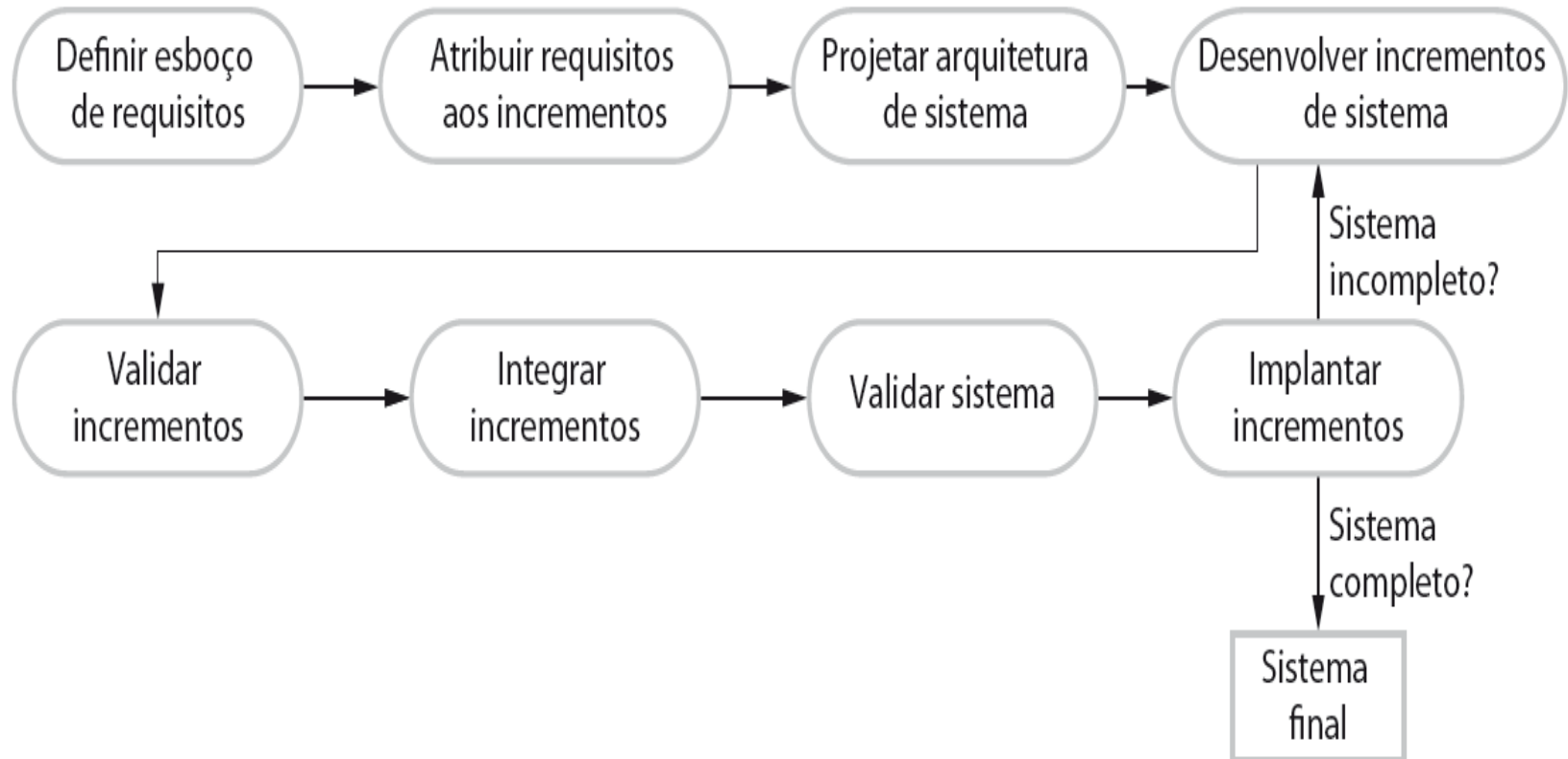
Benefícios do desenvolvimento incremental

- O custo para acomodar mudanças nos requisitos do cliente é reduzido.
 - ✓ A quantidade de análise e documentação que precisa ser feita é bem menor do que o necessária no modelo cascata.
- É mais fácil obter feedback do cliente sobre o trabalho de desenvolvimento que tem sido feito.
 - ✓ Os clientes podem comentar demonstrações do software e ver quanto foi implementado.
- Possibilidade de mais rapidez na entrega e implantação de software útil para o cliente.
 - ✓ Os clientes podem usar e obter ganhos do software mais cedo do que é possível no processo cascata.

Problemas do desenvolvimento incremental

- O processo não é visível.
 - ✓ Gerentes precisam de entregas regulares para medir o progresso. Se os sistemas são desenvolvidos de forma rápida, não é viável do ponto de vista do custo produzir documentação para refletir todas as versões do sistema.
- A estrutura do sistema tende a degradar conforme novos incrementos são adicionados.
 - ✓ A menos que tempo e dinheiro sejam gastos na reconstrução para melhorar o software, as mudanças regulares tendem a corromper a estrutura do sistema. A incorporação posterior de mudanças no software se torna progressivamente mais difícil e cara.

Entrega incremental



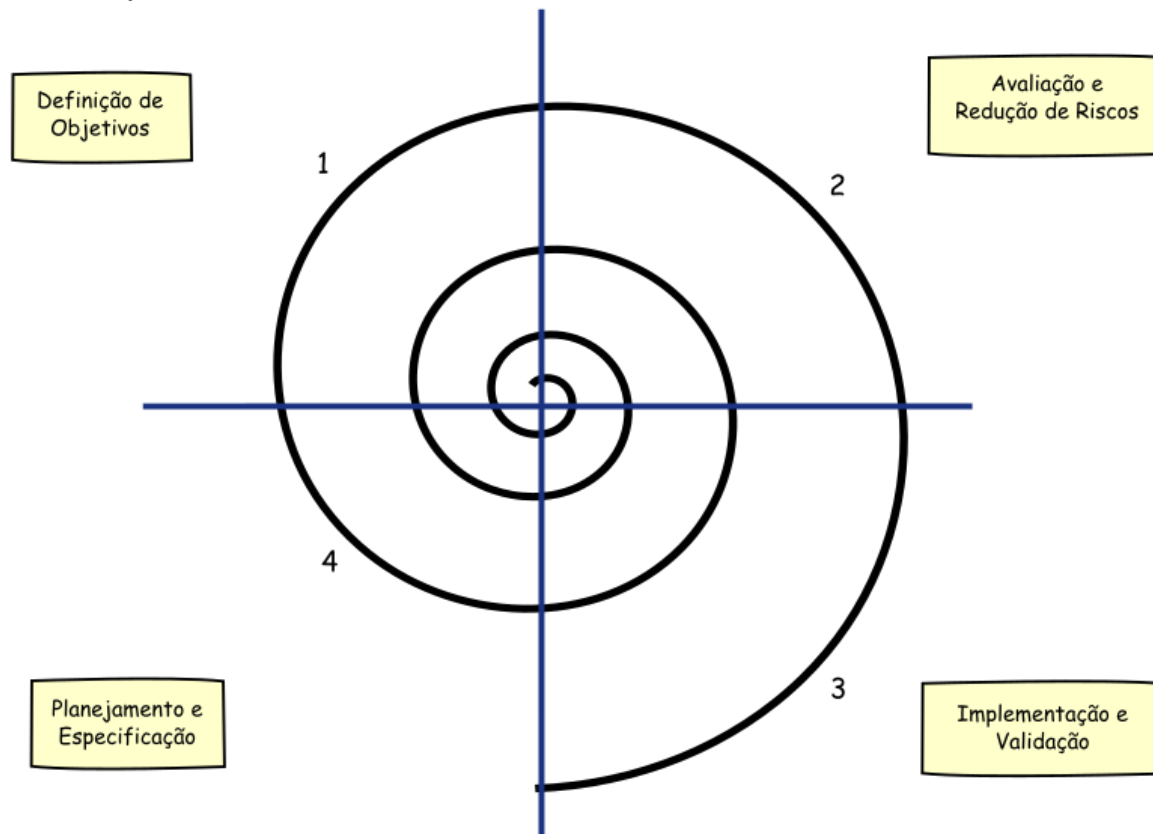
Modelo Espiral

Modelo Espiral

- O processo é representado como uma espiral ao invés de uma sequência de atividades com retornos.
- Cada loop na espiral representa uma fase do processo.
- Não existem fases fixas como especificação ou projeto – os loops na espiral são escolhidos de acordo com a necessidade.
- Os riscos são avaliados explicitamente e resolvidos no decorrer do processo.

Modelo Espiral

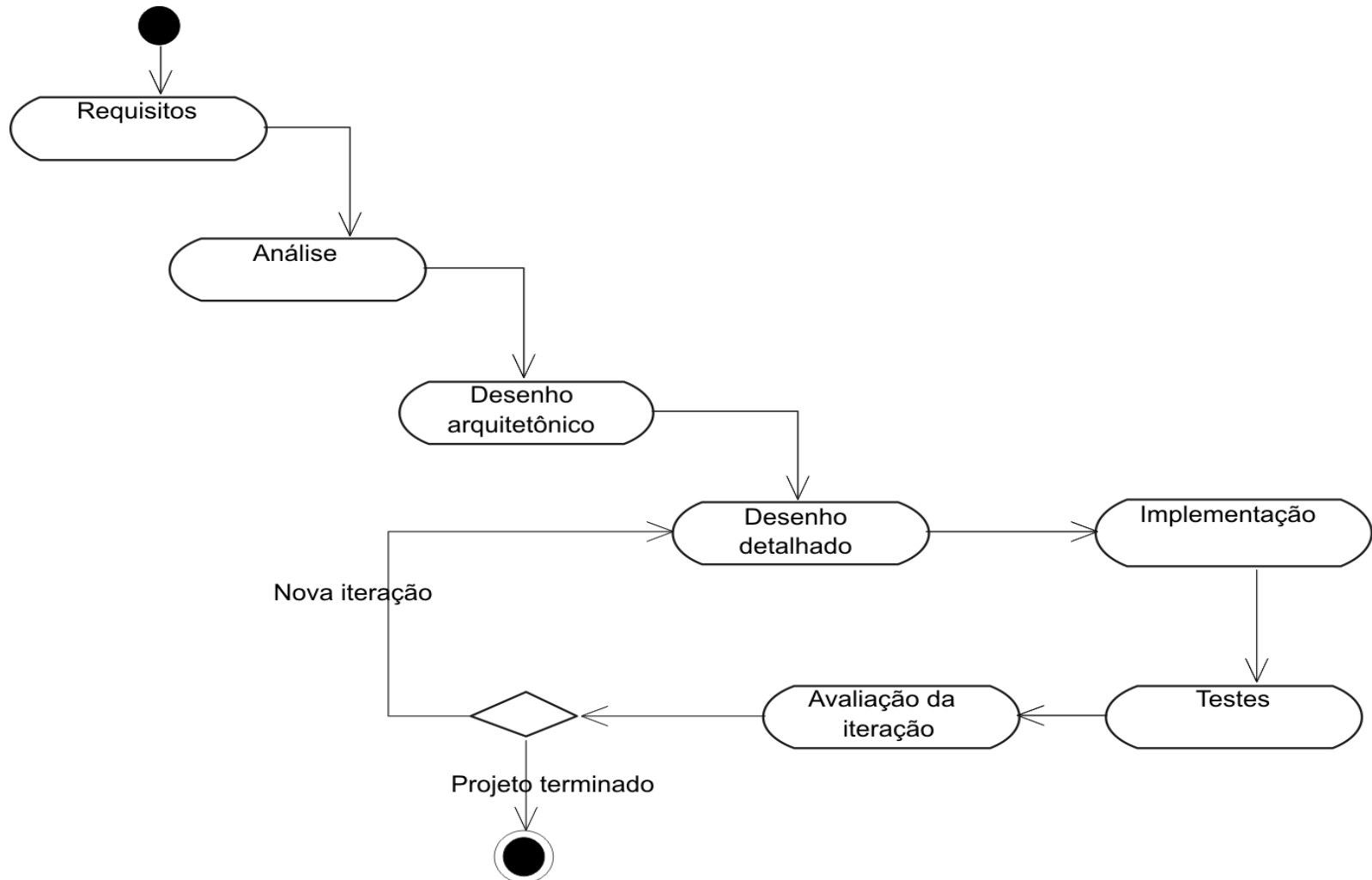
- Na parte mais interna relaciona-se o início da visão da viabilidade do sistema.
- E a cada ciclo, vai evoluindo a visibilidade do sistema como um todo.



Setores do modelo espiral

| SETORES | Descrição |
|--------------------------|--|
| ATIVACÃO | Define-se os objetivos específicos, identifica-se as restrições para o processo e é preparado um plano de gerenciamento detalhado. Identifica-se também os riscos sem analisá-los profundamente (foco da próxima fase). |
| ANÁLISE de RISCOS | Com base nos riscos identificados na fase anterior são realizadas análises detalhadas, e tomadas providências para amenizar esses riscos. Cria-se várias versões de protótipos para apoiar essa fase. |
| DESENVOLVIMENTO | Fundamentado pelas fases anteriores, escolhe-se o modelo mais adequado para o desenvolvimento do Sistema. A bagagem profissional e a vivência do desenvolvedor em outros sistemas, são estratégicas para essa fase. Dependendo da complexidade do Sistema, às vezes, é necessária a presença de um consultor especialista. |
| PLANEJAMENTO | O projeto é revisto nessa fase, e é tomada uma decisão de realizar um novo ciclo na espiral ou não. Se continuar com o aperfeiçoamento do Sistema, é traçado um plano para a próxima fase do projeto. |

Entrega evolutiva



Até a próxima aula...