



Universidade Estadual do Piauí - UESPI
Campus Alexandre Alves de Oliveira - Parnaíba/PI
Bacharelado em Ciência da Computação



Modelos de Processo de Software

Átila Rabelo Lopes
Engenharia de Software

Introdução

- Desenvolvimento de software é uma tarefa complexa
- Depende de vários fatores:
 - Produto a ser desenvolvido
 - Equipe
 - Recursos disponíveis
- Não existe um processo universal e ideal para o desenvolvimento de todos os tipos de softwares

Introdução

- Habilidades necessárias:
 - Criatividade
 - Entendimento do domínio
 - Conhecimento de programação
 - Tomada de decisão
 - Trabalho em equipe
 - Dedicação

Processo de software

Conjunto de atividades relacionadas que levam à produção de um produto de software (*Sommerville, 2013*)

Incluem:

1. Produtos gerados em cada atividade
2. Papéis envolvidos
3. Pré e pós-condições das atividades

Modelo de Processo de Software

- Representação estruturada do conjunto de atividades necessárias para o desenvolvimento e a manutenção de um software.
- Serve como guia para a organização e a execução dessas atividades
 - Garante que o produto final atenda às expectativas e requisitos do cliente.
- Incluem fases de planejamento, design, implementação, teste, e manutenção.
- Auxilia a equipe:
 - Planejar, gerenciar e controlar o desenvolvimento
 - Atender os requisitos
 - Cumprir prazos e orçamento

Modelo de Processo de Software

- Existem vários modelos de processo de software (paradigmas)
- Cada modelo voltado para um determinado contexto
 - Prazo
 - Qualidade
 - Gerenciamento
 - Evolução do software...

Modelos de Processo

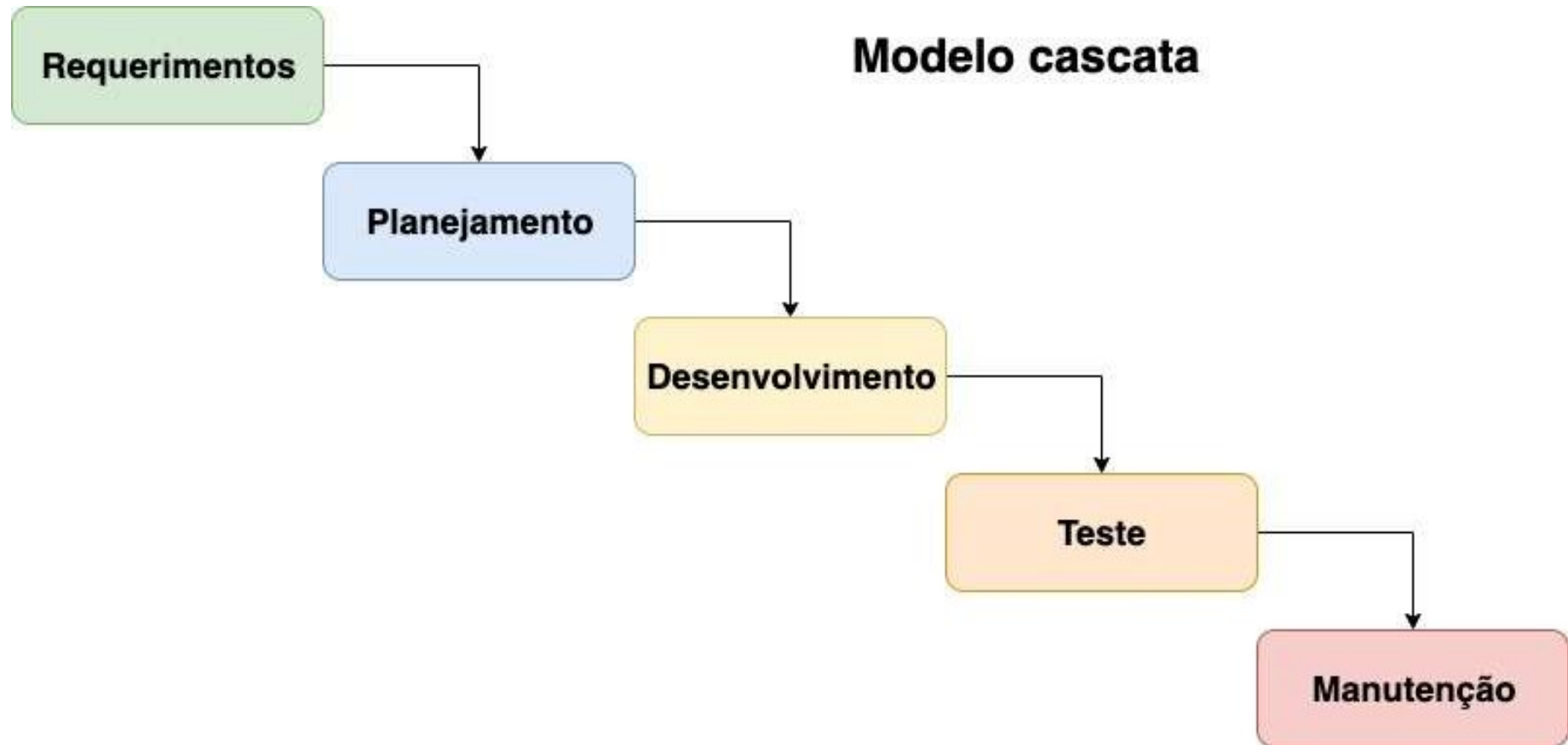
- Modelo Cascata (Sequencial Linear)
- Modelo de Prototipação
- Modelo Evolutivo
 - Modelo Incremental
 - Modelo Espiral
 - Modelo Ágil
- Técnicas de Quarta Geração

Modelo Cascata

Modelo Cascata

- Um dos primeiros modelos de processo
- Adota uma abordagem linear e seqüencial
- Cada fase deve ser concluída antes que a próxima comece.
 - Resultado de uma fase → Entrada da fase seguinte
- Fases:
 - Requisitos → Design → Implementação → Teste → Manutenção.

Modelo Cascata



Modelo Cascata - Requisitos

- Coleta dos requisitos básicos (entender as necessidades do cliente)
- Coleta dos requisitos mais intensificada
 - Compreender o domínio, funções, desempenho e interfaces exigidos
- Validação dos requisitos junto ao cliente
- Documento de requisitos

Modelo Cascata - Projeto

- Design/projeto
- Tradução dos requisitos para representações gráficas
- Modelagem do software
- Arquitetura do sistema

Modelo Cascata - Desenvolvimento

- Codificação/implementação
- Codificação das representações do projeto em uma linguagem de programação
 - Código-fonte
 - Construção do Banco de Dados / Interfaces..

Modelo Cascata - Testes

- Testar o comportamento do software
- Identificar erros e inconsistências
 - Teste dos aspectos lógicos do software
 - Testes das funcionalidades
- Verificar se a entrada gera os resultados esperados

Modelo Cascata - Manutenção

- Começa após a entrega do software ao cliente
- Etapa em que software sobre alterações/mudanças
 - Correção de erros
 - Adaptação do software ao ambiente de trabalho
 - Evolução do software (novas funcionalidades/serviços)

Modelo Cascata

- Vantagem:
 - Simplicidade e clareza no progresso
- Desvantagem:
 - Difícil de aplicar mudanças após o início

Por que a etapa de manutenção é mais difícil no modelo Cascata?

Como a manutenção ocorre no modelo cascata?

Manutenção no Modelo Cascata

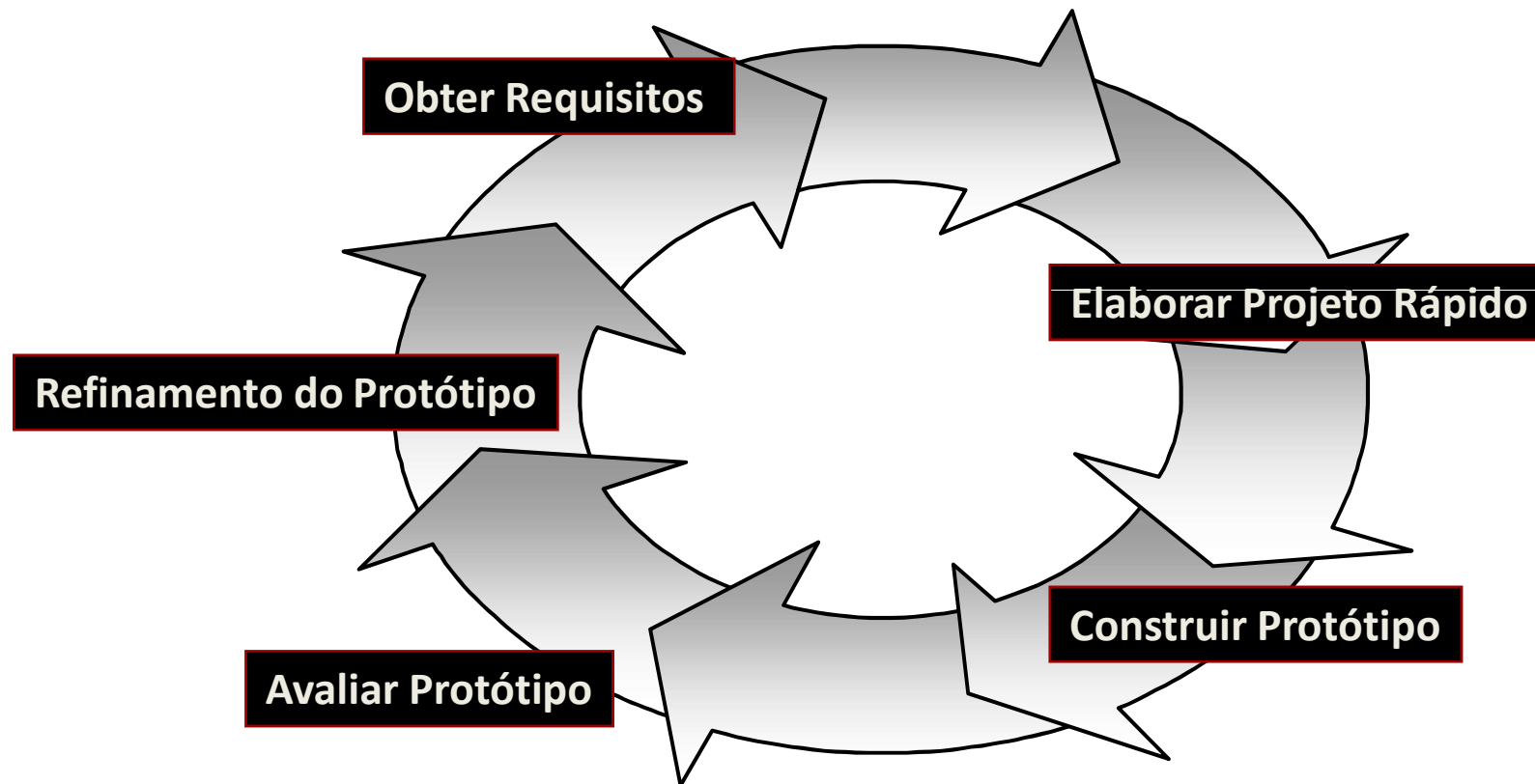
- **Novo ciclo de desenvolvimento**
 - Necessário para corrigir erros ou adicionar novas funcionalidades
 - Dependendo da mudança, pode-se repetir todas as fases do modelo
- **Baixa flexibilidade**
 - Não foi projetado para acomodar mudanças frequentes ou rápidas.
 - O processo pode ficar mais lento e caro em comparação com outros
- **Documentação detalhada:**
 - Ajuda a garantir que as mudanças mantenham o padrão do software

Modelo de Prototipação

Modelo de Prototipação

- Foca na construção rápida da versão inicial ou protótipo do software para que possa ser avaliado pelos usuários.
- O objetivo é entender os requisitos do usuário para obter uma melhor definição dos requisitos do sistema
- Fases:
 1. Levantamento inicial de requisitos.
 2. Construção do protótipo.
 3. Avaliação pelo cliente.
 4. Refinamento do protótipo até chegar ao sistema final.

Modelo de Prototipação



Modelo de Prototipação

1- OBTENÇÃO DOS REQUISITOS:

desenvolvedor e cliente definem os objetivos gerais do software, identificam quais requisitos são conhecidos e as áreas que necessitam de definições adicionais.

Refinamen

o

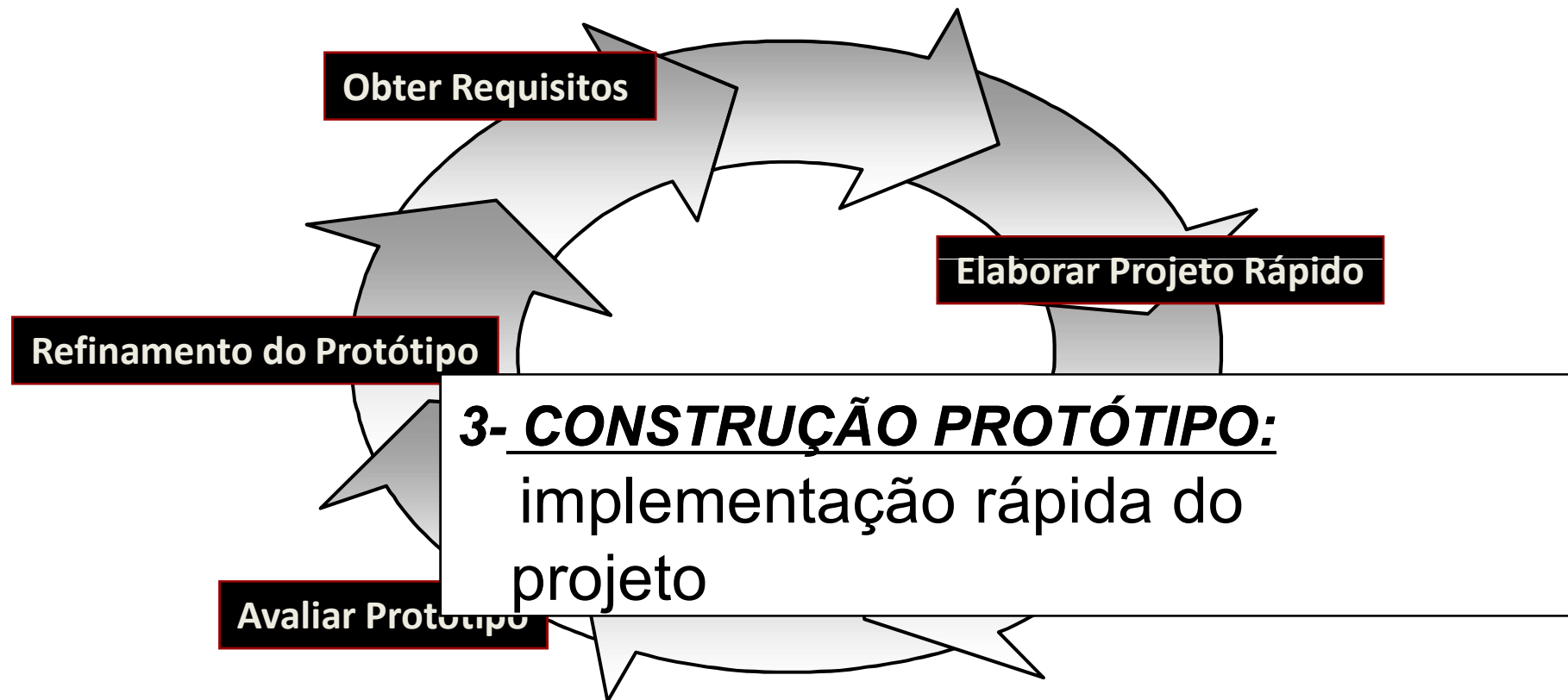
Avaliar Protótipo

Construir Protótipo

Modelo de Prototipação



Modelo de Prototipação



Modelo de Prototipação



Modelo de Prototipação

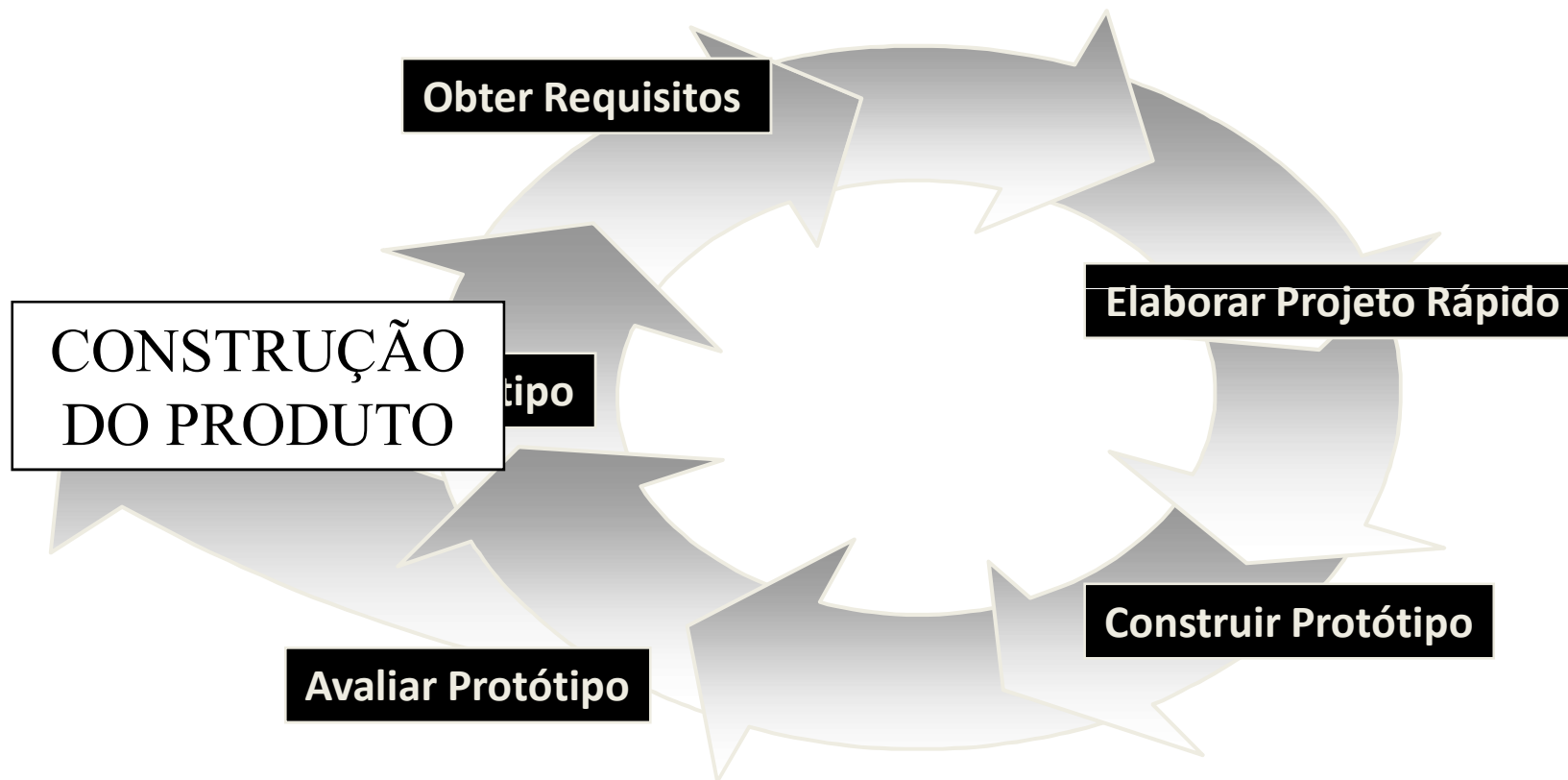


Obter Requisitos

5- REFINAMENTO DO PROTÓTIPO: cliente e desenvolvedor refinam os requisitos do software a ser desenvolvido.

Avaliar Protótipo

Modelo de Prototipação



Modelo de Prototipação

Obter Requisitos

6- CONSTRUÇÃO PRODUTO: identificados os requisitos, o protótipo deve ser descartado e a versão de produção deve ser construída considerando os critérios de qualidade.

Objeto Rápido

Protótipo

Avaliar Protótipo

Modelo de Prototipação

- Vantagens:
 - Facilita a compreensão dos requisitos.
 - Reduz o risco da má comunicação entre desenvolvedores e clientes.
- Desvantagens:
 - Protótipos podem ser confundidos com o sistema final.
 - Se não gerenciado corretamente, pode resultar em um processo mais demorado.

Modelo Evolutivo

Modelo Evolutivo

- São modelos utilizados no desenvolvimento de softwares que precisam evoluir com o passar do tempo
- Consiste no desenvolvimento de forma incremental, permitindo melhorias e adaptações contínuas conforme a demanda
- Modelo iterativo
- Permite o desenvolvimento de versões cada vez mais completas

Modelo Evolutivo

Objetivo: Prover a evolução do software por meio do desenvolvimento gradativo

- Fases:
 1. Definição dos requisitos básicos.
 2. Desenvolvimento da primeira versão.
 3. Avaliação e feedback.
 4. Refinamento e adição de novas funcionalidades.
- Vantagens:
 - Facilita a adaptação a mudanças de requisitos.
 - Feedback contínuo do cliente.
- Desvantagens:
 - Pode ser difícil gerenciar um projeto em constante mudança.
 - A documentação pode ficar defasada.

Modelo Evolutivo

- Necessidades de evolução do software:
 - Mudanças nos requisitos do software e/ou modelo de negócio, durante o desenvolvimento
 - Prazo curto para a entrega do software completo
 - Quando os requisitos importantes já são conhecidos, mas ainda falta definir alguns detalhes

Modelo Evolutivo

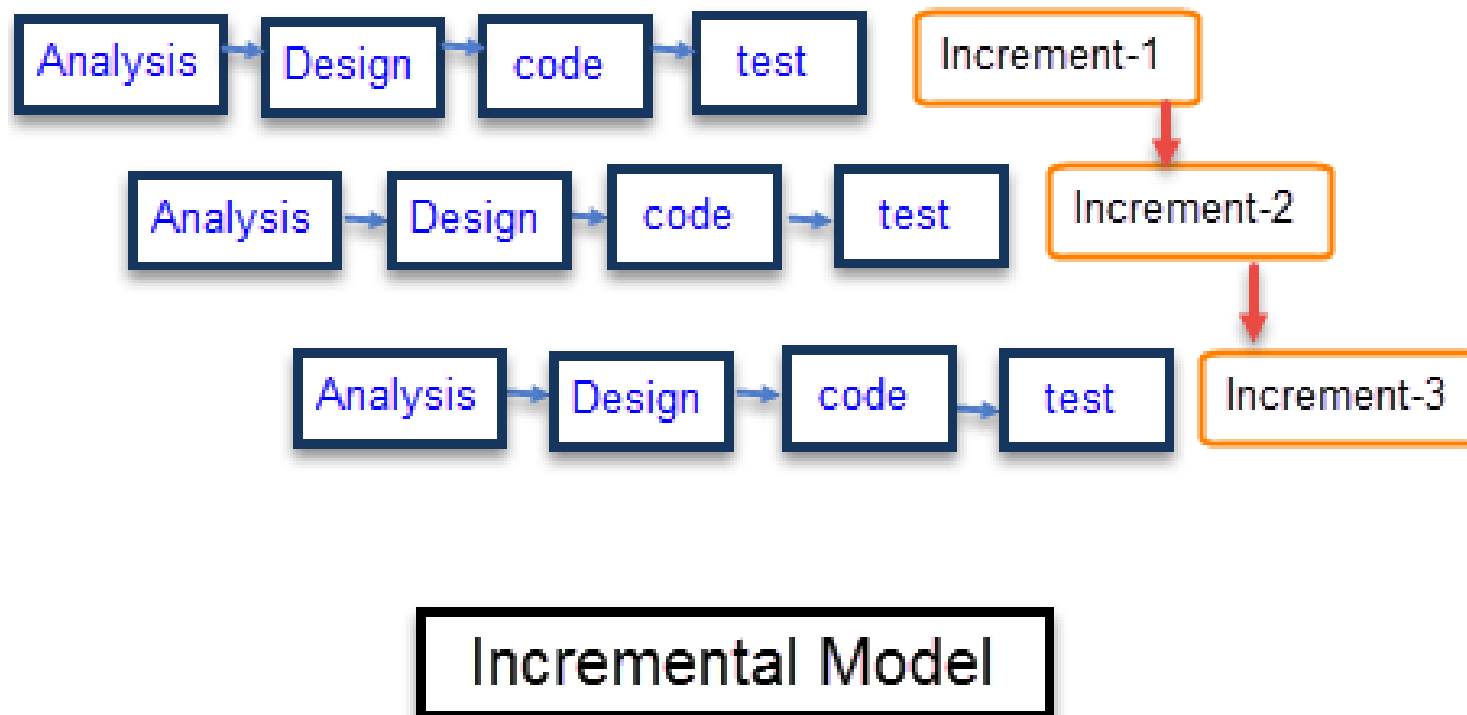
- Tipos de Modelos Evolutivos:
 - **Modelo Incremental**
 - **Modelo Espiral**
 - **Modelo Ágil**
 - Modelo de Montagem de Componentes
 - Modelo de Desenvolvimento Concorrente
 - Processo Unificado

Modelo Incremental

Modelo Incremental

- O desenvolvimento é dividido em pequenas partes (Incrementos)
 - Cada parte é projetada, desenvolvida e testada de maneira independente
- Objetivo:
 - Entregar o sistema em partes, onde cada parte agrega valor ao sistema final.
- Fases:
 1. Análise/Requisitos
 2. Projeto/Design
 3. Codificação
 4. Teste

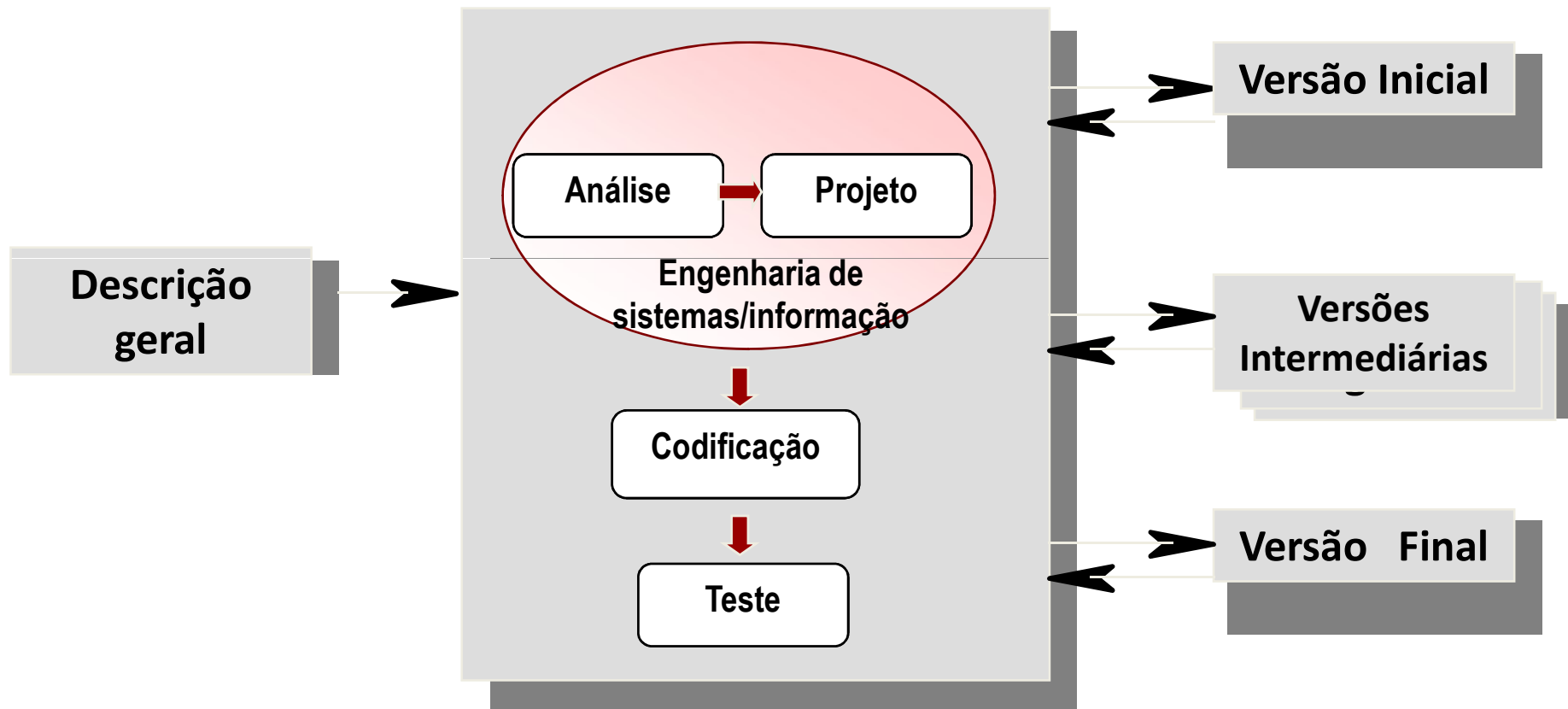
Modelo Incremental



Modelo Incremental

- O primeiro incremento entregue é o produto principal (requisitos básicos)
- Após validado pelo cliente, novos incrementos vão sendo integrados no produto principal
- O processo se repete até a conclusão do software
- Vantagens:
 - Entregas parciais permitem testes e validações contínuas.
 - Requisitos podem ser ajustados entre incrementos.
- Desvantagens:
 - Integração contínua pode ser complexa.
 - Demora para o cliente receber o software completo
 - Requisitos não tão bem definidos inicialmente podem causar atrasos.

Modelo Incremental



Modelo Espiral

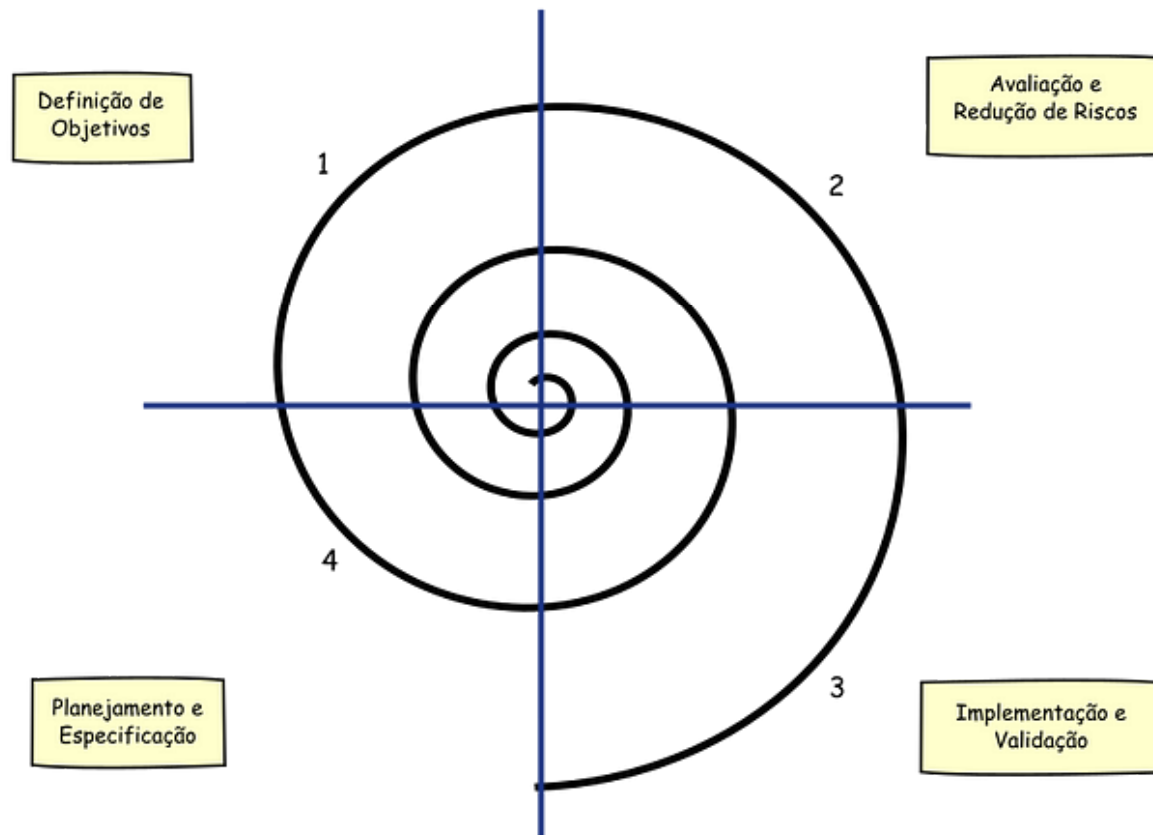
Modelo Espiral

- Incorpora aspectos do modelo incremental, cascata e prototipação, incluindo uma nova abordagem (Análise de riscos)
- O desenvolvimento ocorre em vários ciclos,
 - Cada ciclo representa uma etapa de desenvolvimento
- Os riscos vão sendo minimizados a cada nova iteração
- Usa a Prototipação em qualquer etapa da evolução do produto

Fases:

1. Definição de objetivos
2. Avaliação e análise de riscos
3. Desenvolvimento e validação (primeira iteração)
4. Planejamento da próxima iteração

Modelo Espiral



Modelo Espiral

Definição dos objetivos

- Definição dos objetivos específicos
- Identificação de restrições do projeto e produto
- Elaboração do plano de gerenciamento
- Identificação dos riscos do projeto

Avaliação e redução de riscos

- Análise detalhada dos riscos identificados
- Planejamento e execução das ações para reduzir os riscos

Ex: Se houver risco de existirem requisitos inadequados, pode-se desenvolver protótipos do sistema para verificar a importância dos requisitos

Modelo Espiral

Desenvolvimento e validação

- Após a avaliação de risco, um modelo de desenvolvimento de sistema deve ser selecionado:
 - Se os riscos de interfaces forem os principais riscos identificados pode-se adotar o modelo de prototipação
 - Se o risco de integração de subsistemas for o principal risco identificado pode-se escolher o modelo em cascata

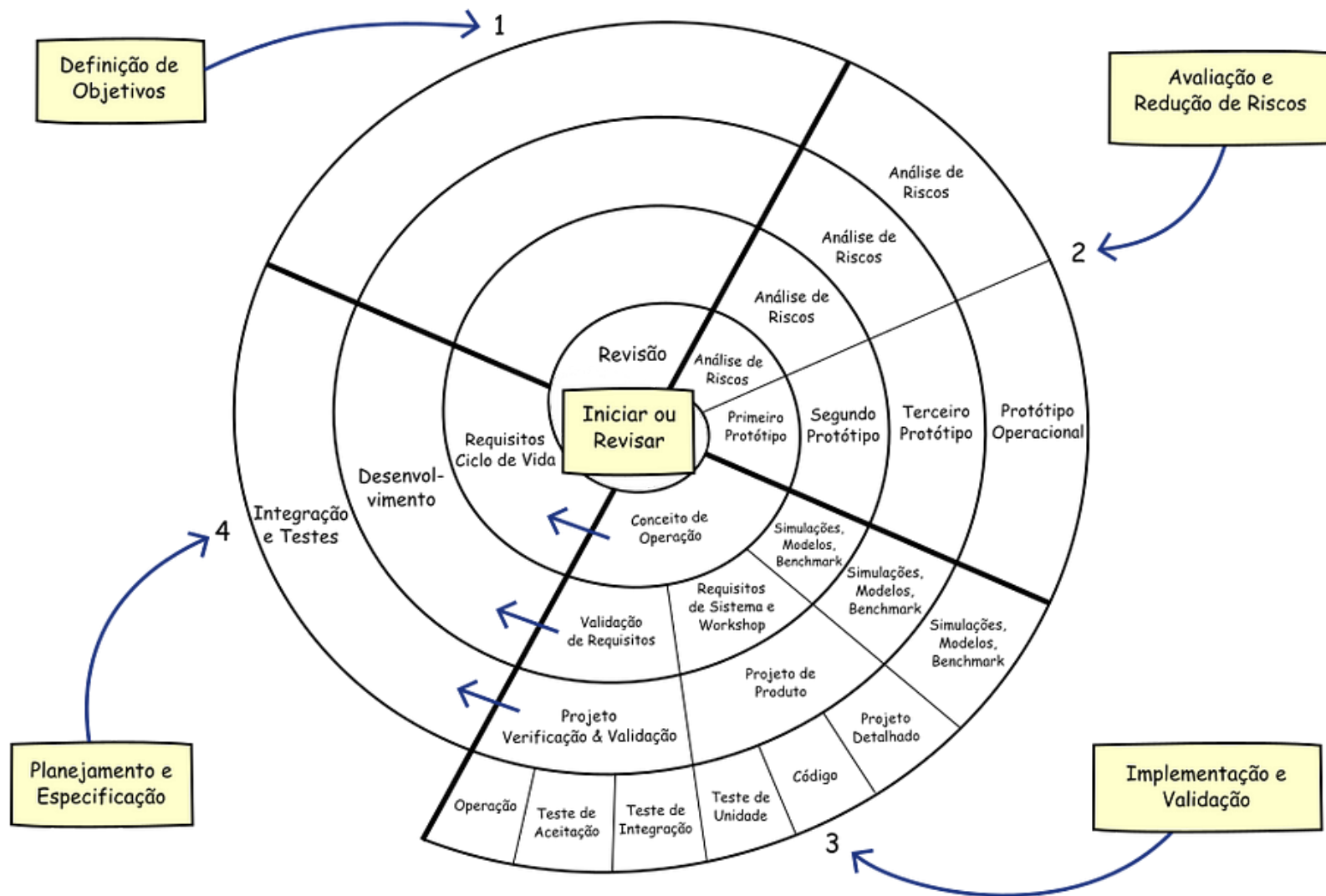
Planejamento e especificação

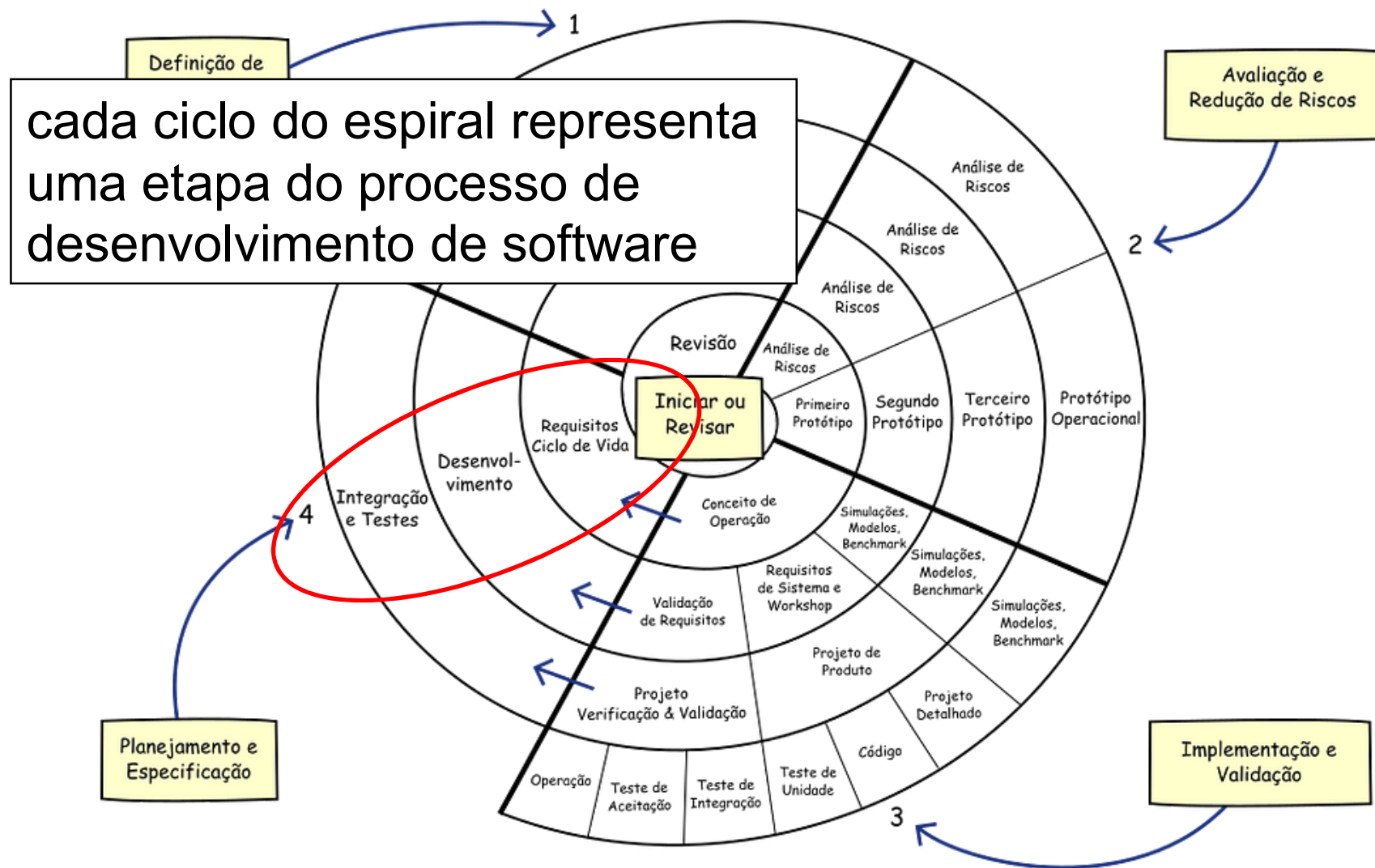
- O projeto é revisado e decidido se passa para o próximo ciclo
- Planejamento e especificação das atividades da próxima fase do projeto

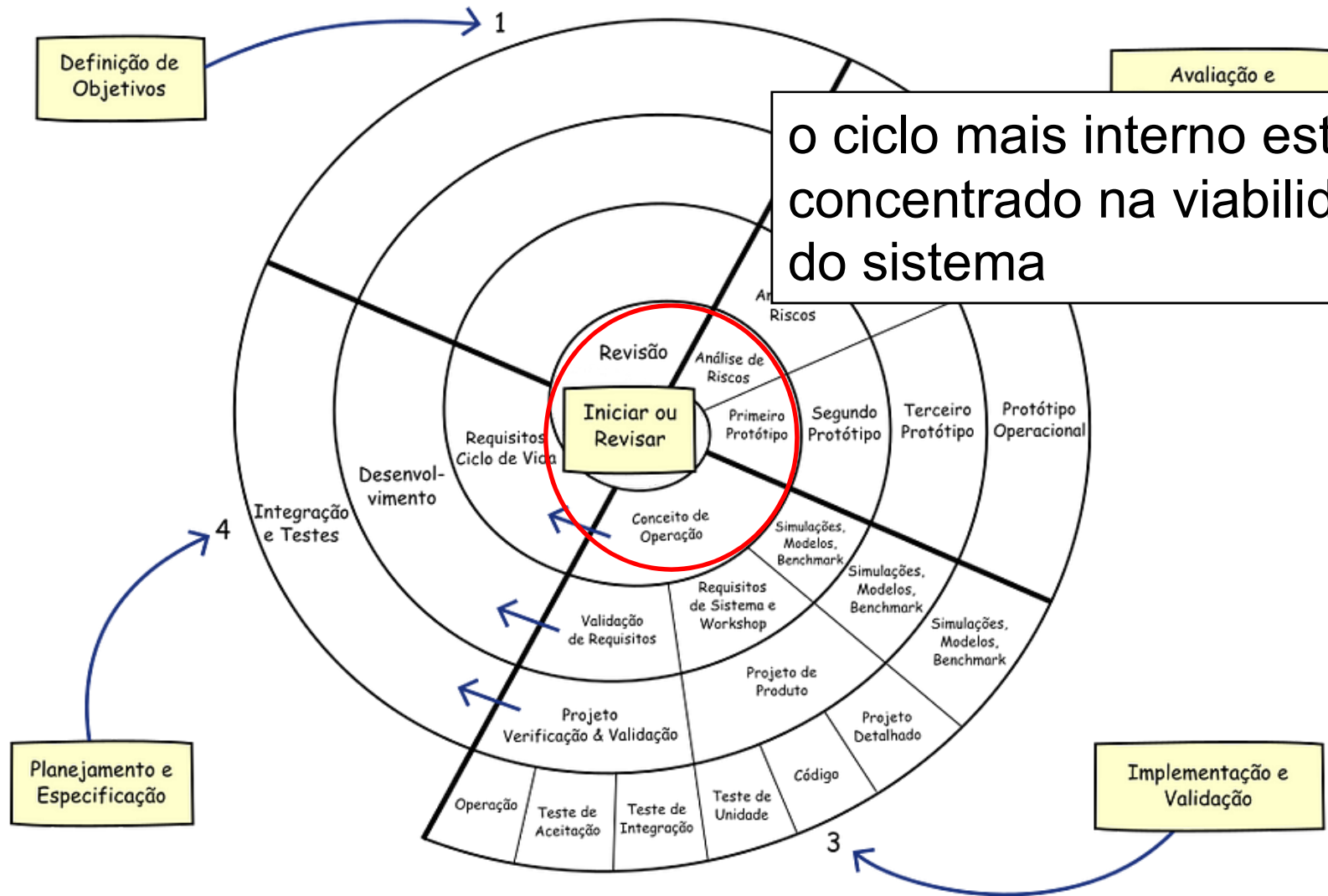
Modelo Espiral

Cada ciclo do espiral envolve:

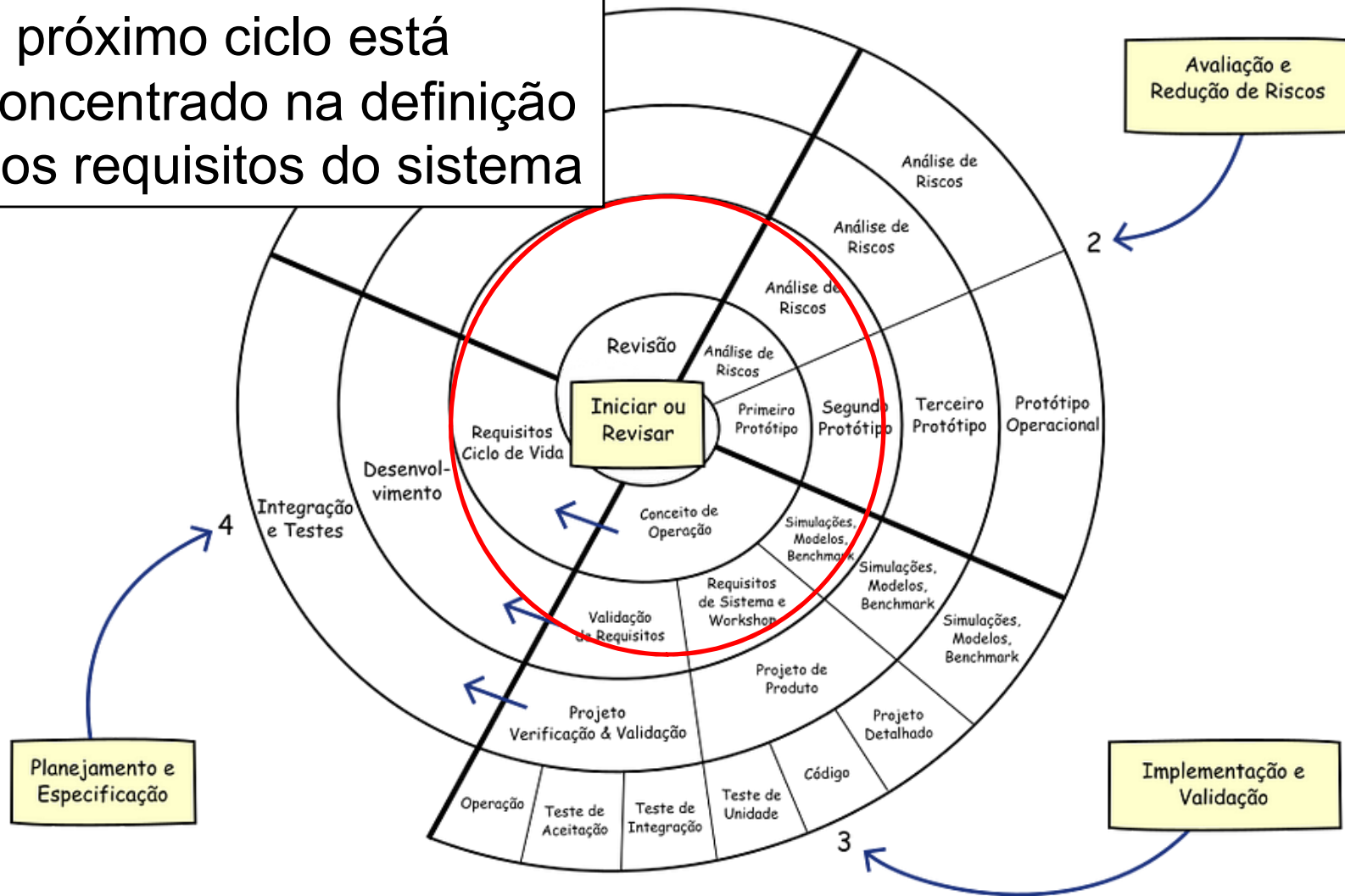
- Determinar os objetivos, alternativas e restrições relacionadas à iteração que vai se iniciar
- Identificar e resolver os riscos relacionados
- Avaliar alternativas disponíveis. Podem ser feitos protótipos para analisar a viabilidade de diferentes alternativas
- Desenvolver os artefatos relacionados à iteração corrente e valida-los
- Planejar a próxima iteração
- Obter concordância em relação ao planejamento

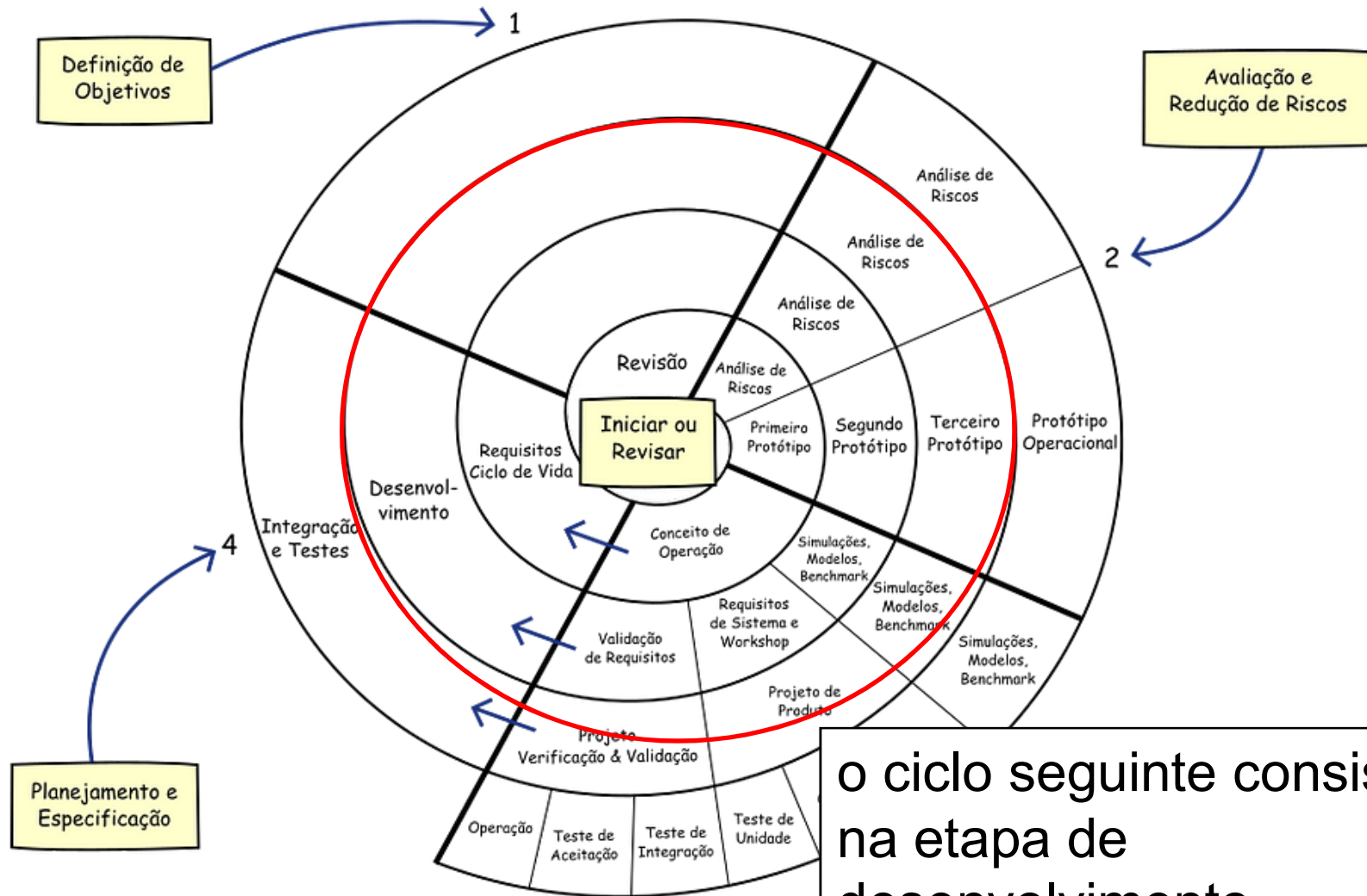


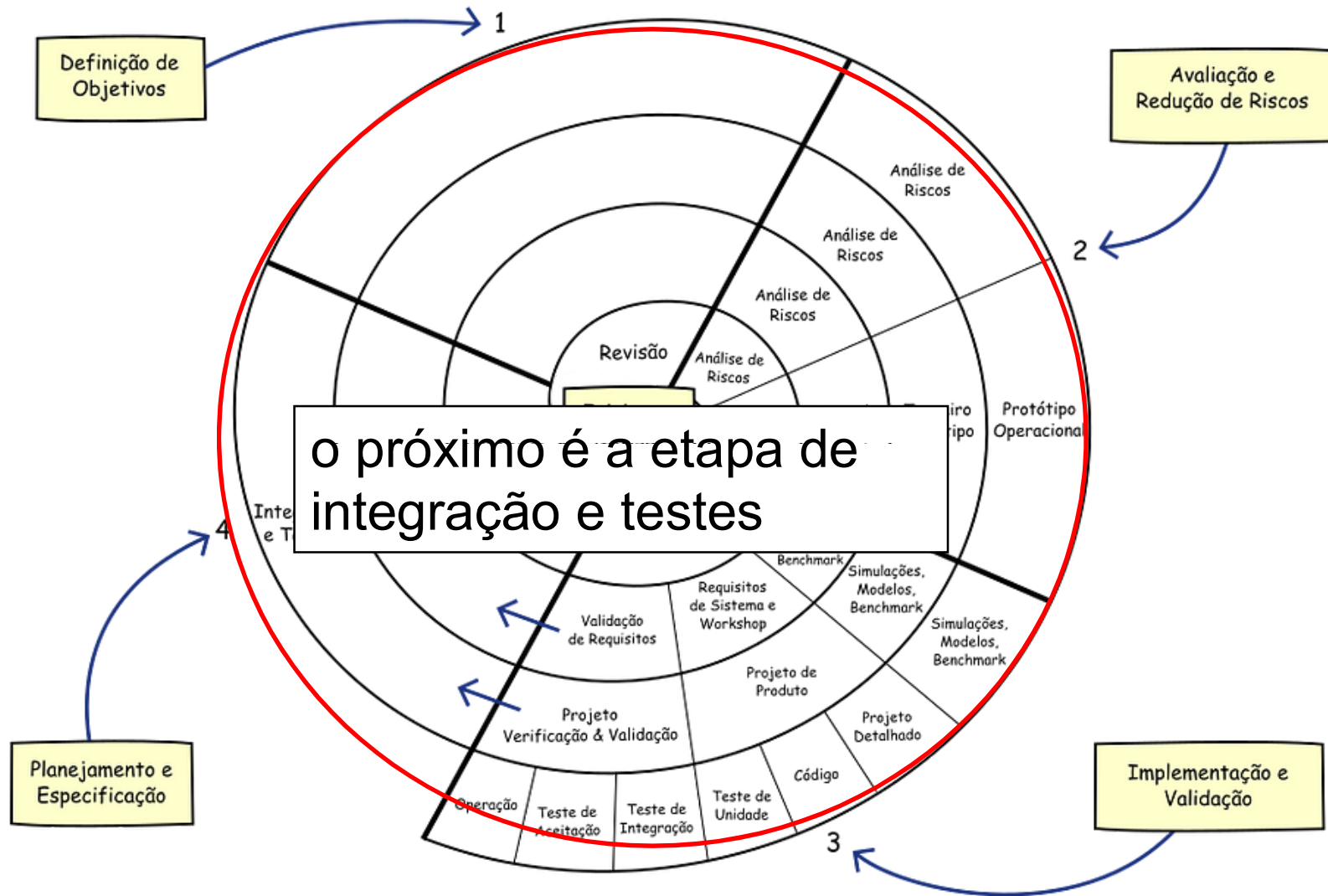


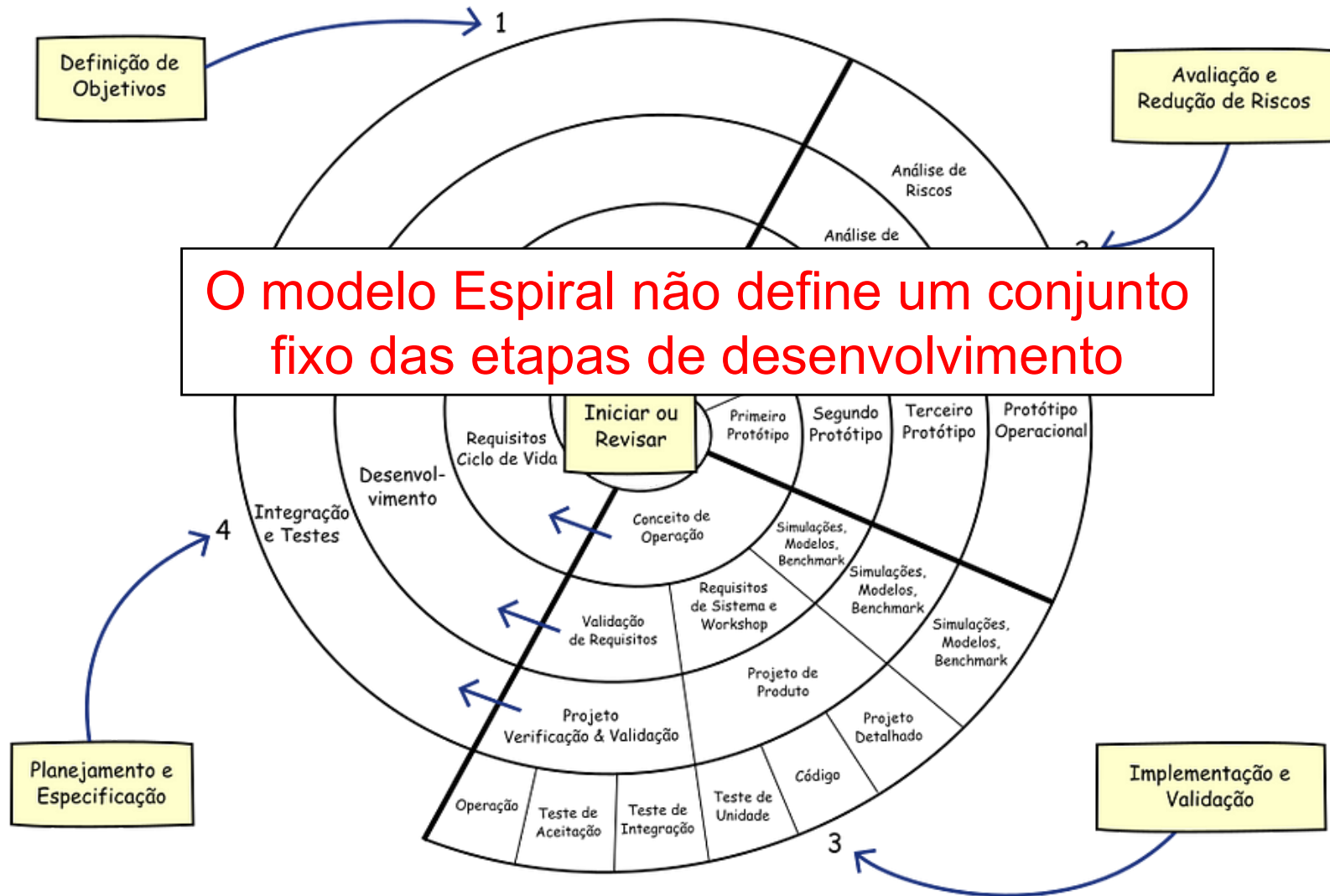


o próximo ciclo está
concentrado na definição
dos requisitos do sistema









Modelo Espiral

Vantagens

- Adequado para projetos grandes e complexos
- Gerenciamento contínuo dos riscos
- Versátil a mudanças
- Engloba características de outros modelos

Desvantagens

- Mais difícil de gerenciar
- Exige experiência dos analistas de risco
- Pode ser mais caro e demorado (quantidade de iterações)