

Modelos de Processo de Software II – continuação

Seiji Isotani, Rafaela V. Rocha

sisotani@icmc.usp.br
rafaela.vilela@gmail.com

PAE: Armando M. Toda

armando.toda@gmail.com

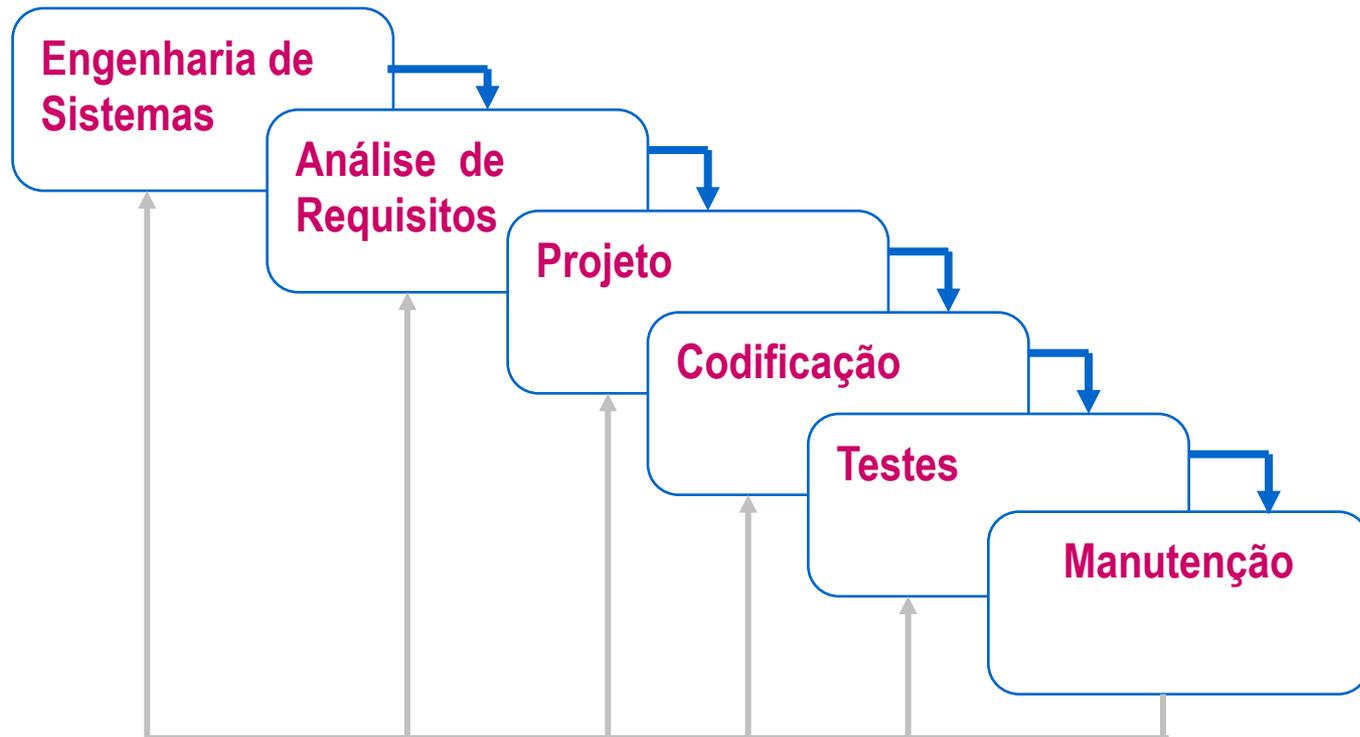


(material produzido e atualizado pelos professores do grupo de pesquisa em Engenharia de Software do ICMC-USP)

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 Ágil*
 - *O Processo Unificado*
 - **O Modelo Espiral**
 - **O Modelo de Montagem de Componentes**
- **Modelo de Métodos Formais**
- **Técnicas de Quarta Geração**

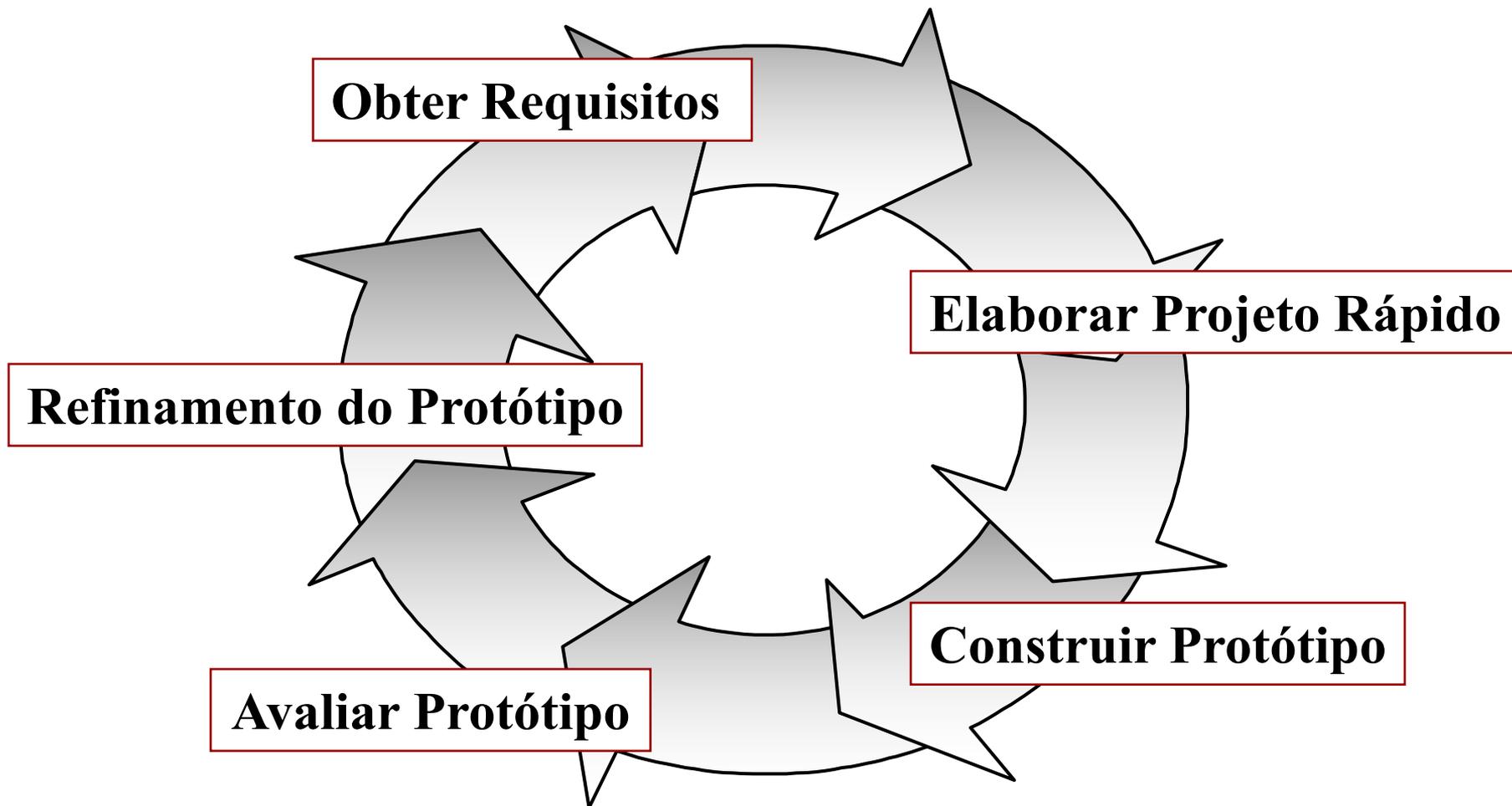
O Modelo Cascata



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 Ágil*
 - *O Processo Unificado*
 - *O Modelo Espiral*
 - *O Modelo de Montagem de Componentes*
- *Modelo de Métodos Formais*
- *Técnicas de Quarta Geração*

O Modelo de Prototipação para obtenção dos requisitos



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 Ágil*
 - *O Processo Unificado*
 - *O Modelo Espiral*
 - *O Modelo de Montagem de Componentes*
- *Modelo de Métodos Formais*
- *Técnicas de Quarta Geração*

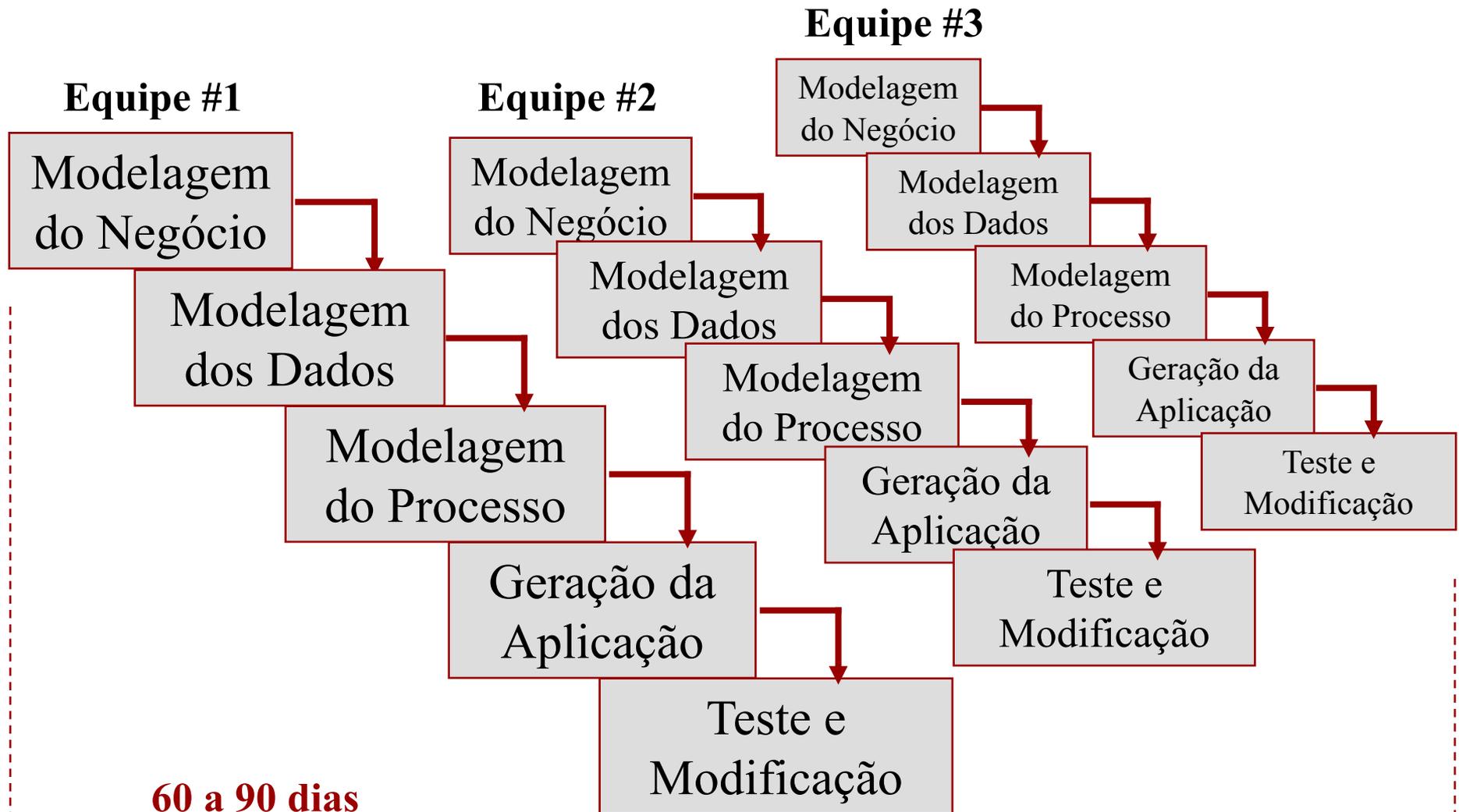
Modelo RAD

- *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**.

O Modelo RAD

- 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.

O Modelo RAD



O Modelo RAD

Desvantagens:

- 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

O Modelo RAD

- Nem todos os tipos de aplicação são apropriadas para o RAD:
 - Deve ser possível a modularização efetiva da aplicação
 - se alto desempenho é uma característica e o desempenho é obtido sintonizando as interfaces dos componentes do sistema, a abordagem RAD pode não funcionar

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 Ágil*
 - *O Processo Unificado*
 - *O Modelo Espiral*
 - *O Modelo de Montagem de Componentes*
 - *O Modelo de Desenvolvimento Concorrente*
- *Modelo de Métodos Formais*
- *Técnicas de Quarta Geração*

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 de Processo

- quando os requisitos de produto e de negócio mudam conforme o desenvolvimento procede
- quando há uma data de entrega apertada (mercado) - impossível a conclusão de um produto completo
- quando um conjunto de requisitos importantes é bem conhecido, porém os detalhes ainda devem ser definidos

Modelos Evolutivos de Processo

- modelos evolutivos são iterativos
- possibilitam o desenvolvimento de versões cada vez mais completas do software

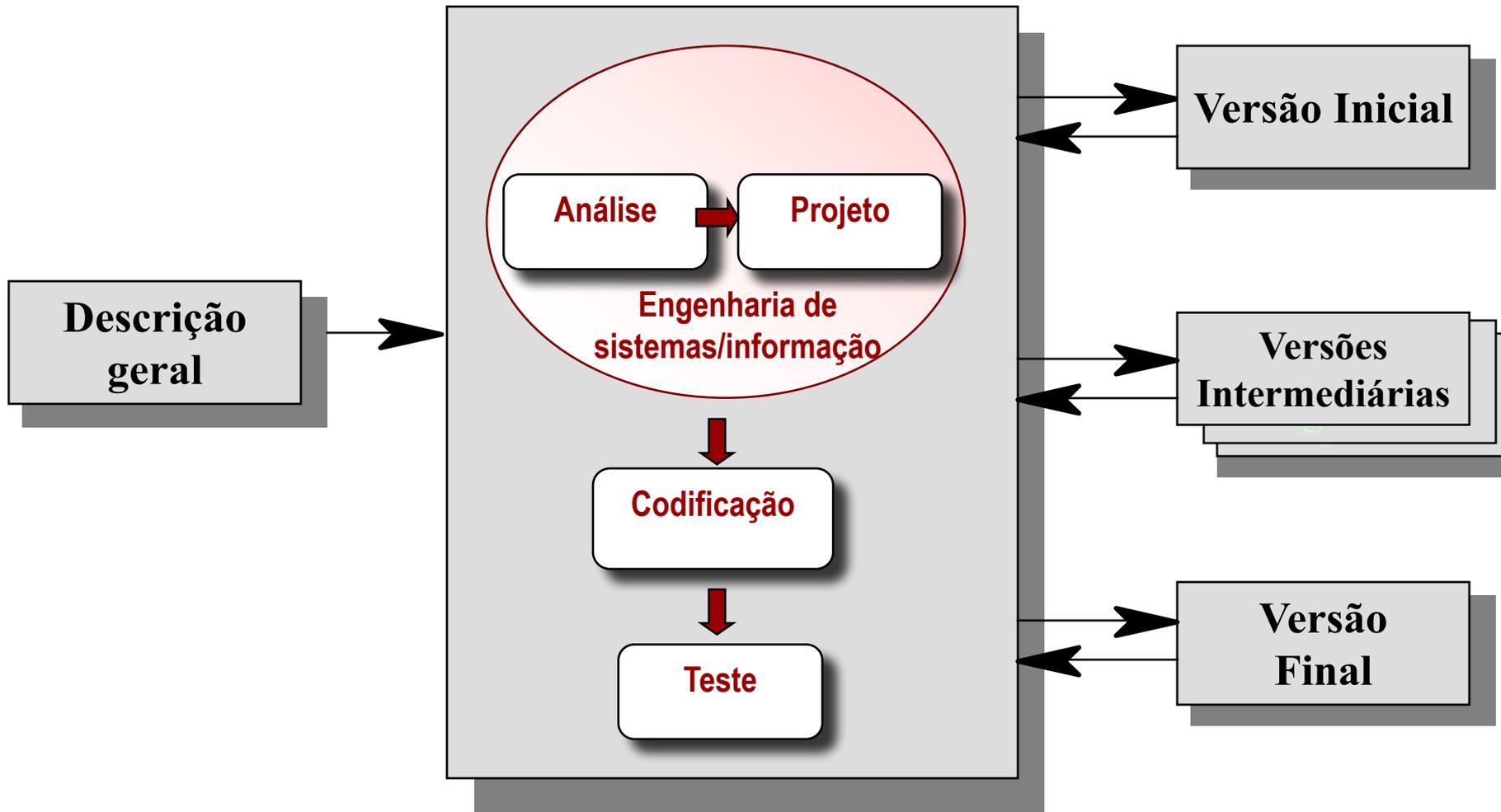
Modelos de Processo de Software

- *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 Ágil*
 - *O Processo Unificado*
 - *O Modelo Espiral*
 - *O Modelo de Montagem de Componentes*
- *Modelo de Métodos Formais*
- *Técnicas de Quarta Geração*

O 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.

O Modelo Incremental



O Modelo Incremental

- a versão inicial é frequentemente o **núcleo** do produto (a parte mais importante)
 - a evolução acontece quando novas características são adicionadas à medida que são sugeridas pelo usuário
- Este modelo é importante quando é difícil estabelecer *a priori* uma especificação detalhada dos requisitos

O Modelo Incremental

- o modelo incremental é mais apropriado para sistemas pequenos
- As novas versões podem ser planejadas de modo que os riscos técnicos possam ser administrados (Ex. disponibilidade de determinado hardware)

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 Ágil*
 - *O Processo Unificado*
 - *O Modelo Espiral*
 - *O Modelo de Montagem de Componentes*
- *Modelo de Métodos Formais*
- *Técnicas de Quarta Geração*

O Modelo Espiral

- Engloba as melhores características do Modelo Cascata e da Prototipação, adicionando um novo elemento: a **Análise de Riscos**.
 - **Risco**: algo que pode dar errado.
 - Segue a abordagem de passos sistemáticos do Modelo Cascata, incorporando-os numa **estrutura iterativa**.
 - É dividido em uma série de regiões, tipicamente de 3 a 6, que delimitam atividades de arcabouço (*framework activities*)
 - Usa a Prototipação em qualquer etapa da evolução do produto, como mecanismo de **redução** de riscos.

Modelo Espiral

- É uma abordagem realista para o desenvolvimento de sistemas e software de **grande porte**.
- Exige a consideração direta dos **riscos técnicos** em todos os estágios do projeto.
 - Se aplicado adequadamente, pode reduzir os riscos antes que eles fiquem problemáticos.

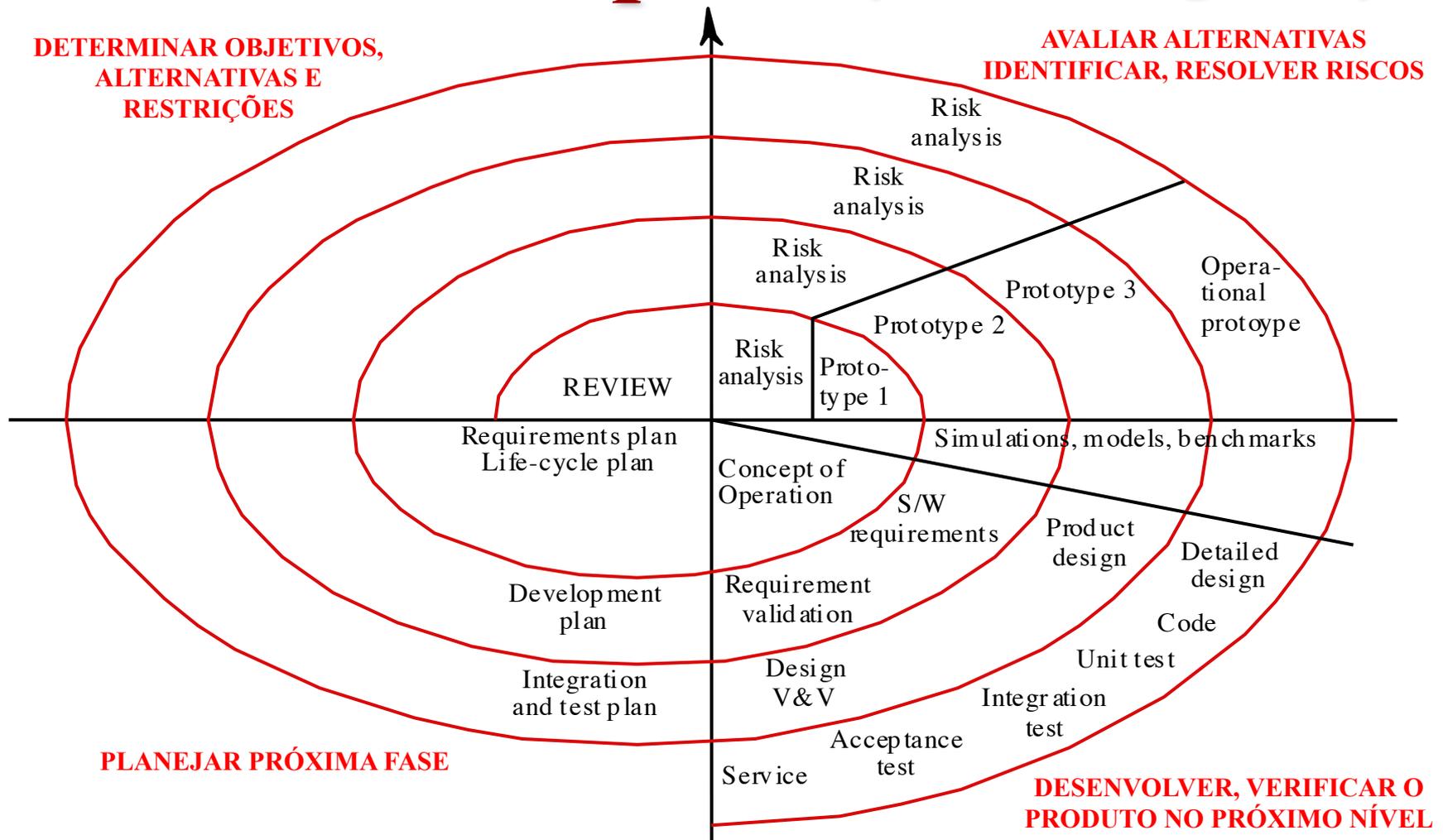
O Modelo Espiral

➤ Funcionamento:

■ Para cada volta na espiral:

- 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

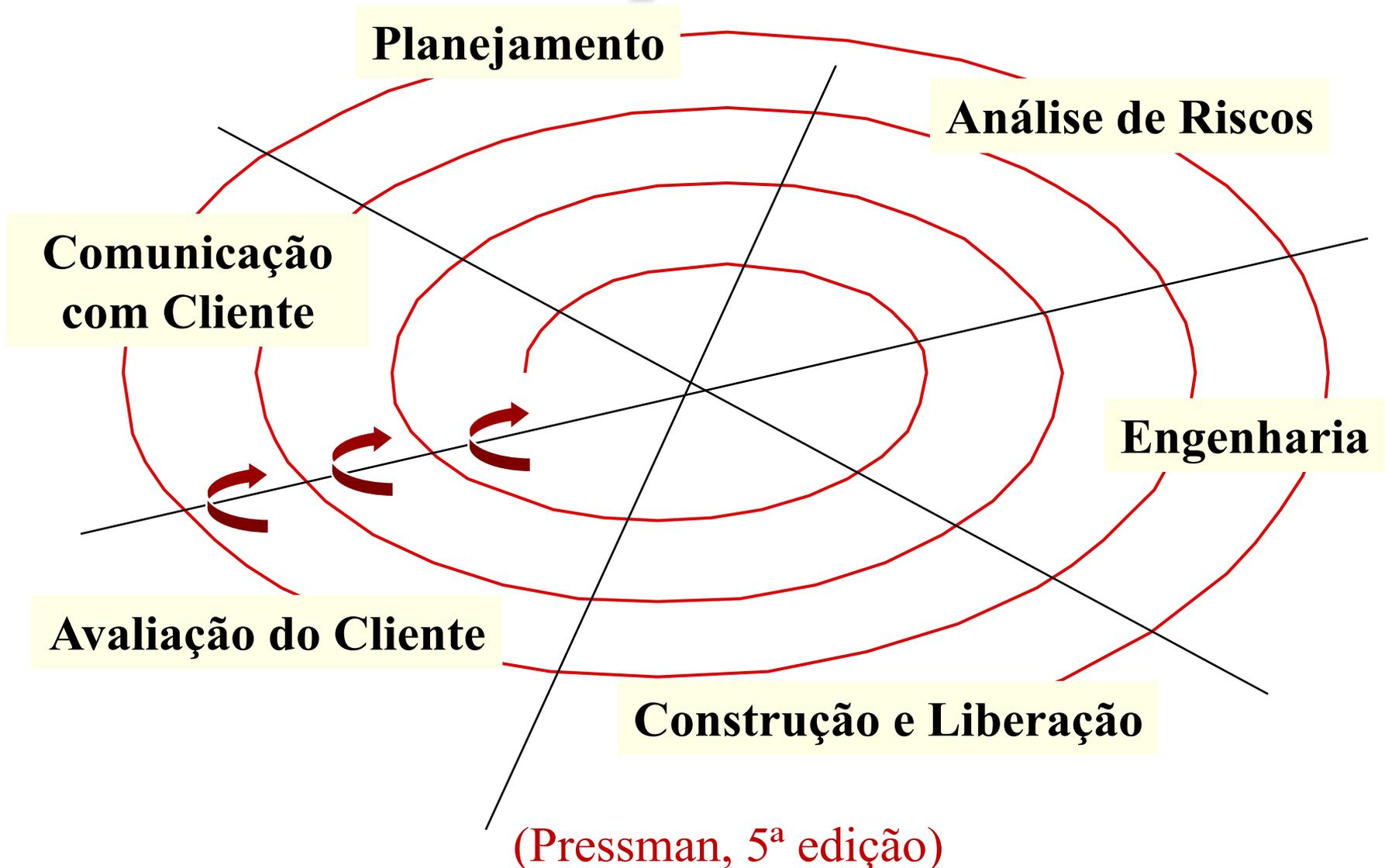
O Modelo Espiral (com 4 regiões)



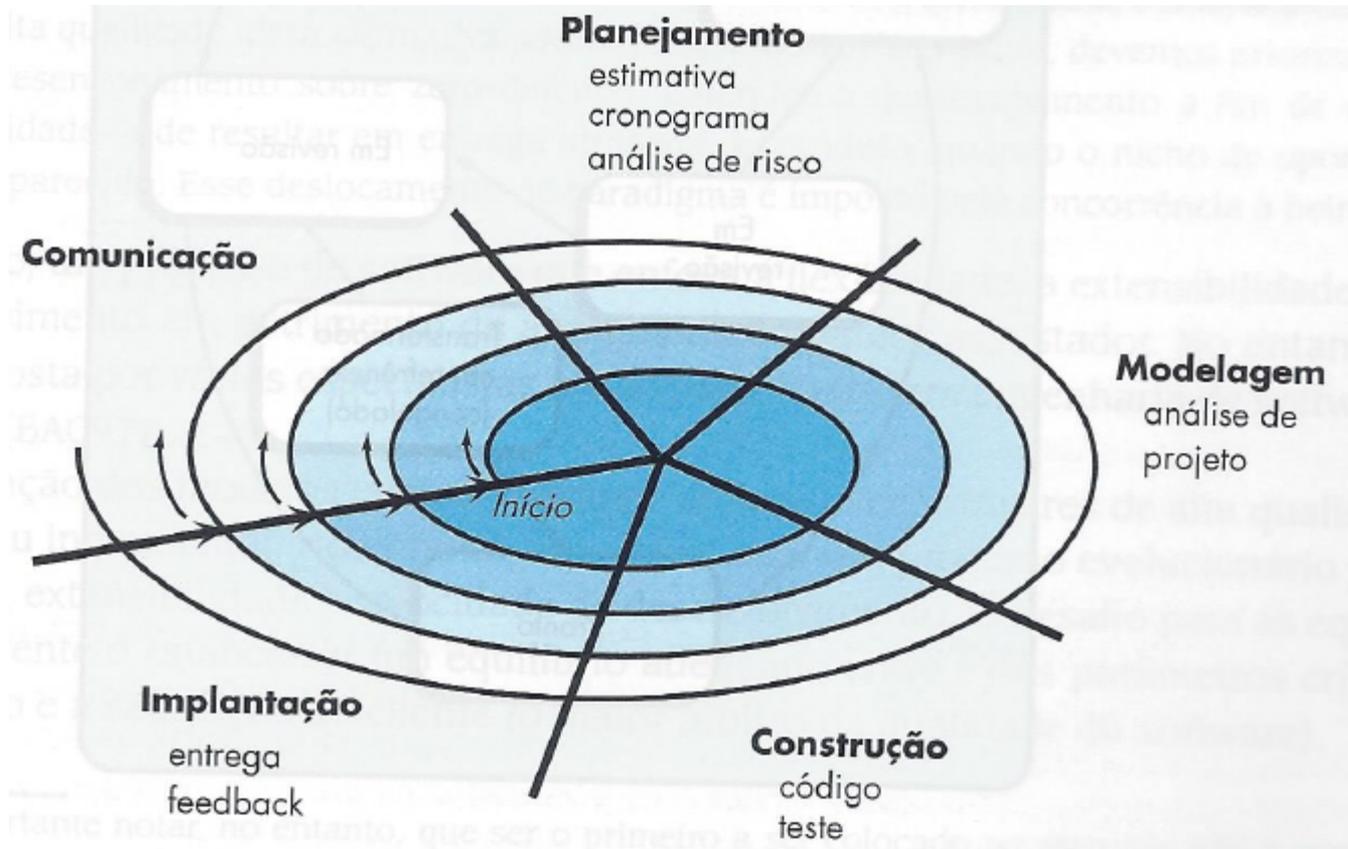
(Sommerville, 9ª edição)

(Wazlawick, 2013)

O Modelo Espiral (com 6 regiões)

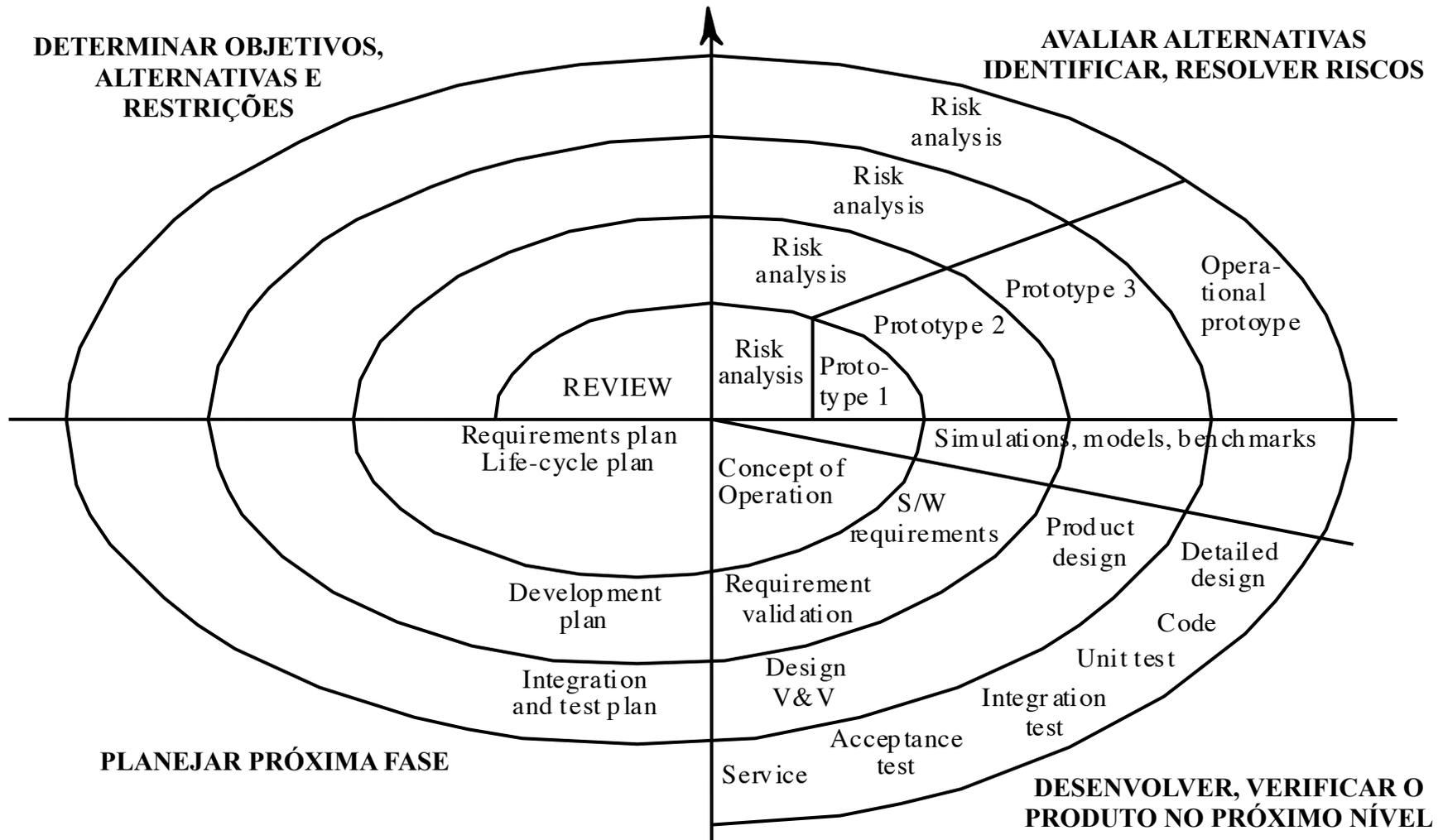


O Modelo Espiral (com 5 regiões)

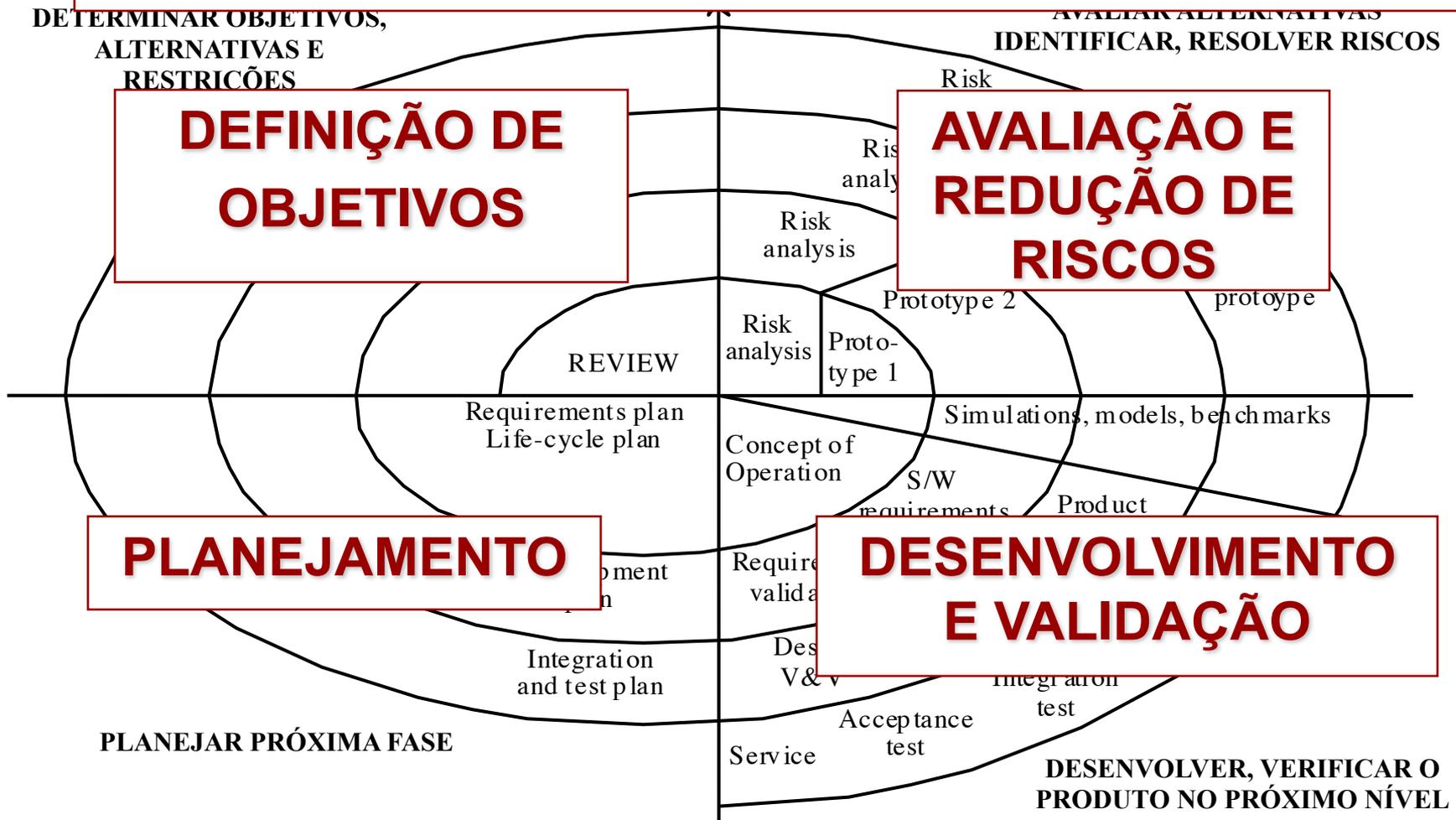


(Pressman, 6ª edição)

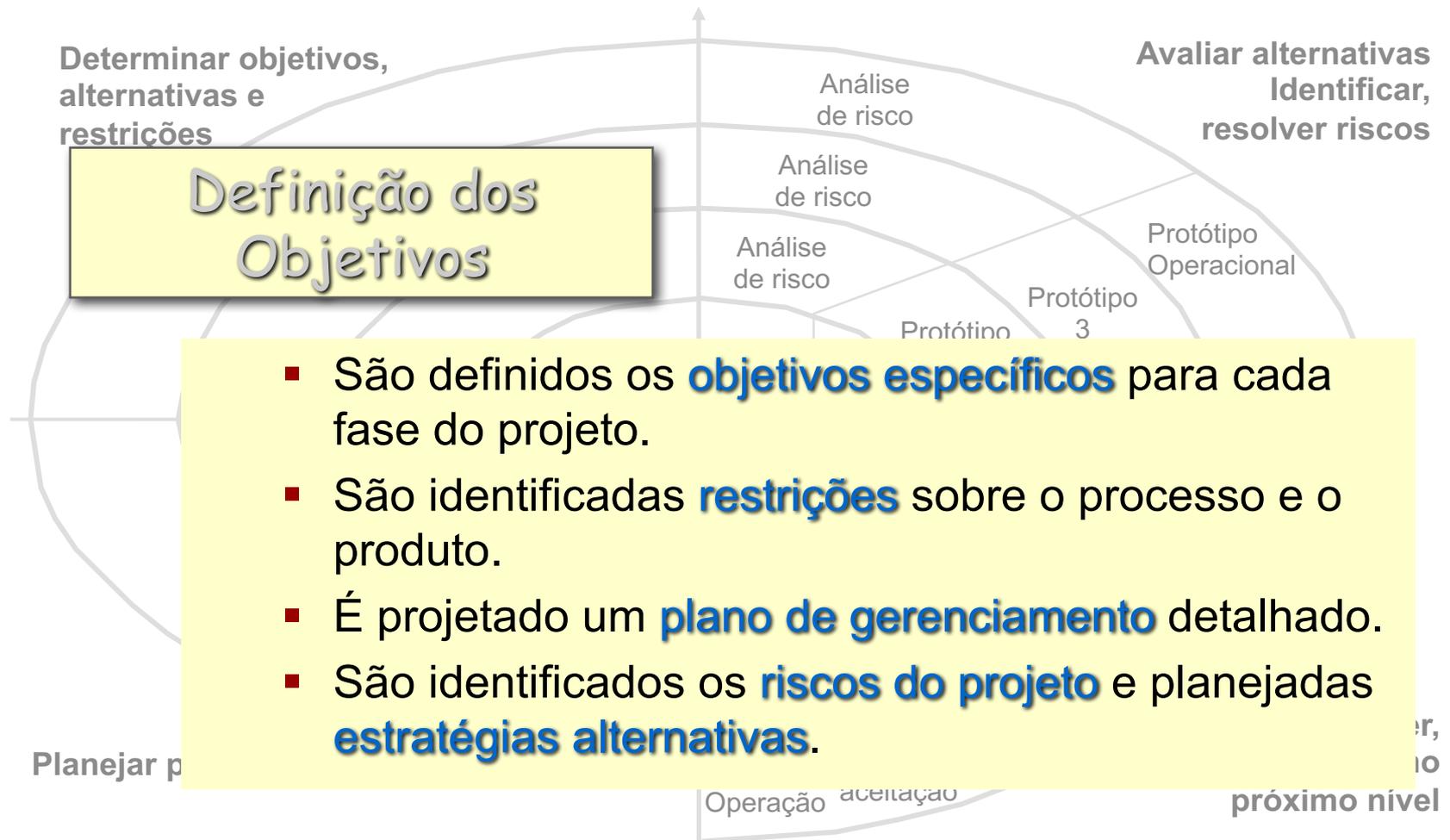
O Modelo Espiral (com 4 regiões)



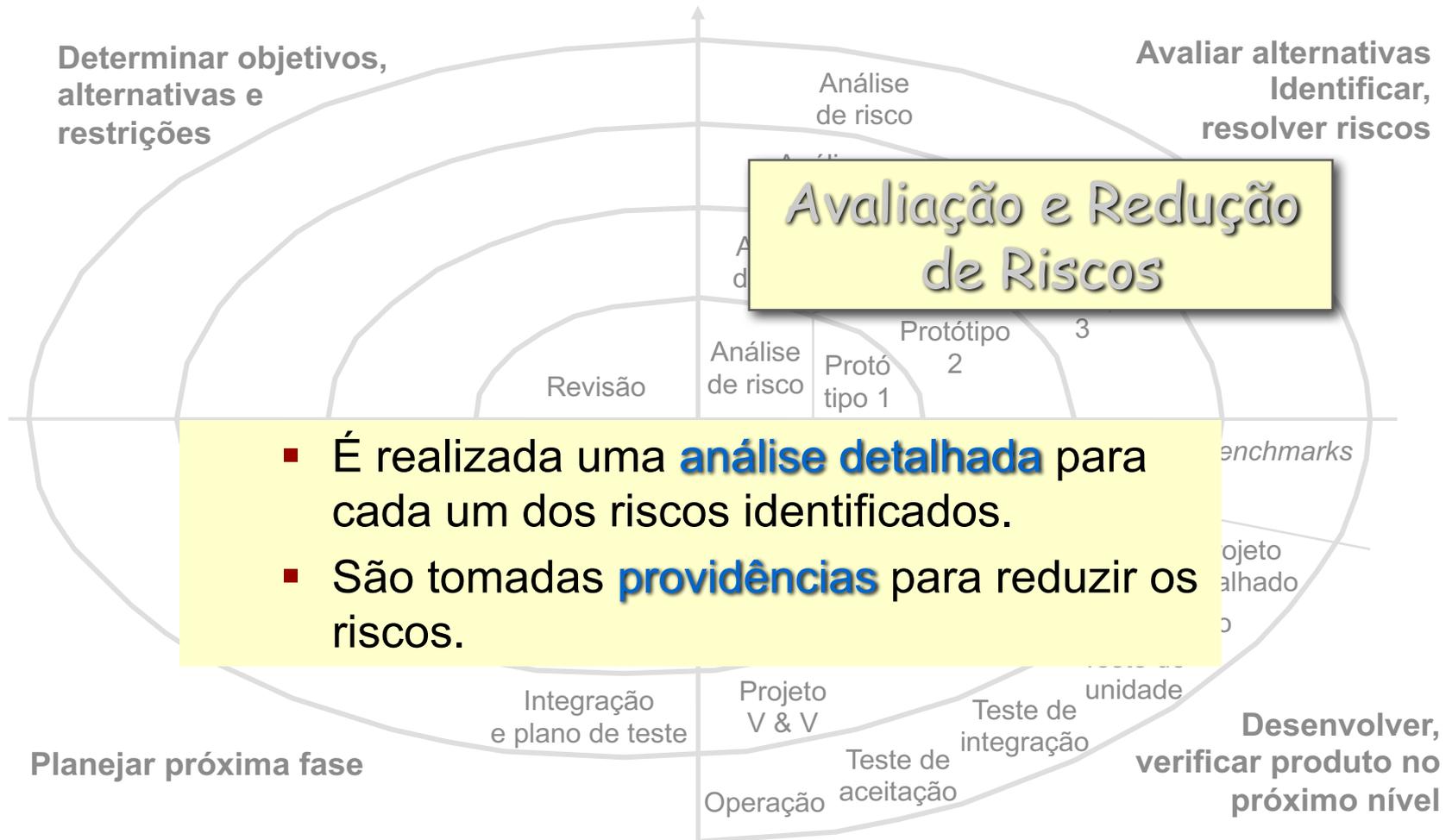
- No modelo com 4 regiões, cada “loop” do espiral é dividido em 4 partes



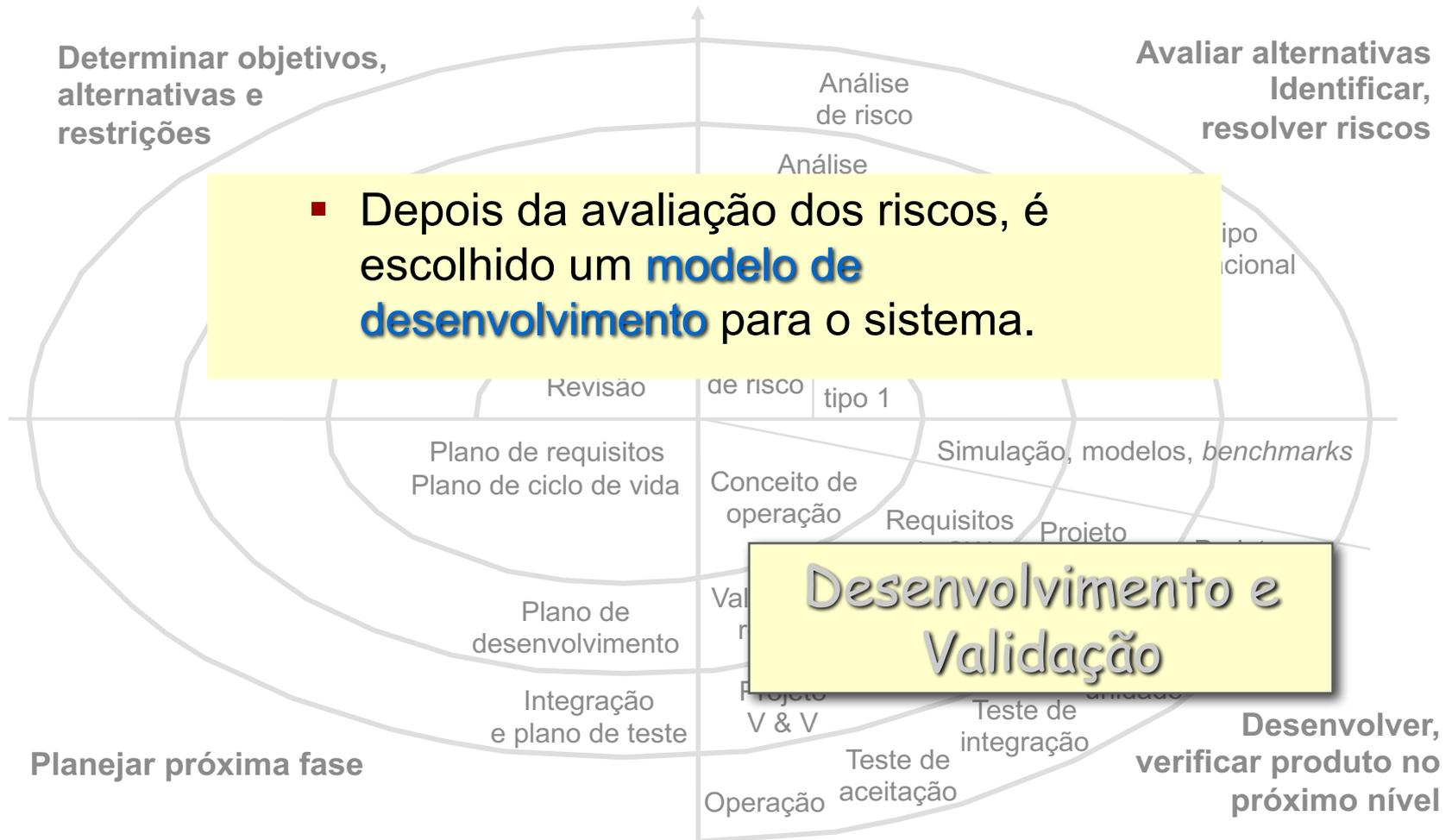
O Modelo Espiral



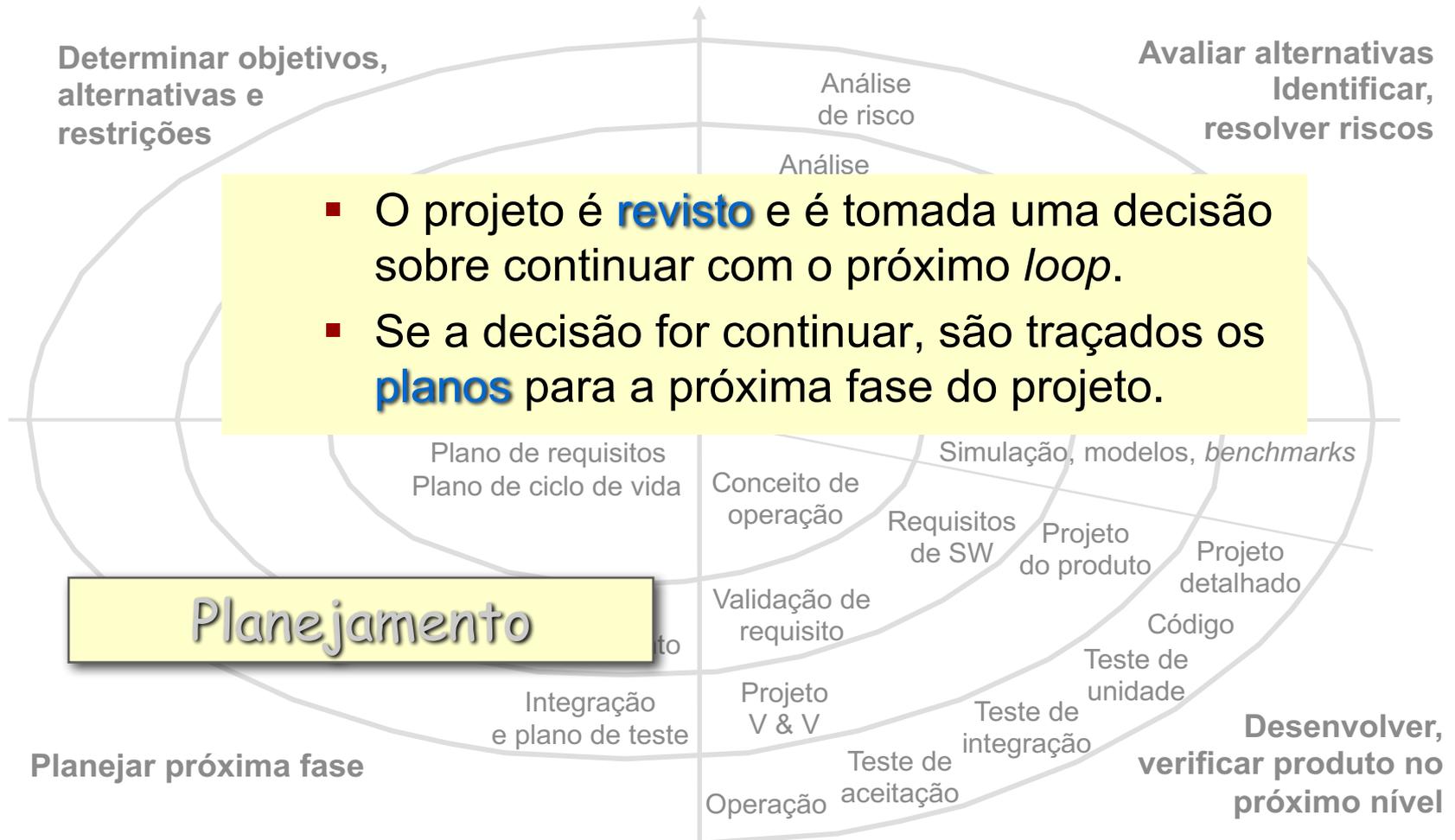
O Modelo Espiral



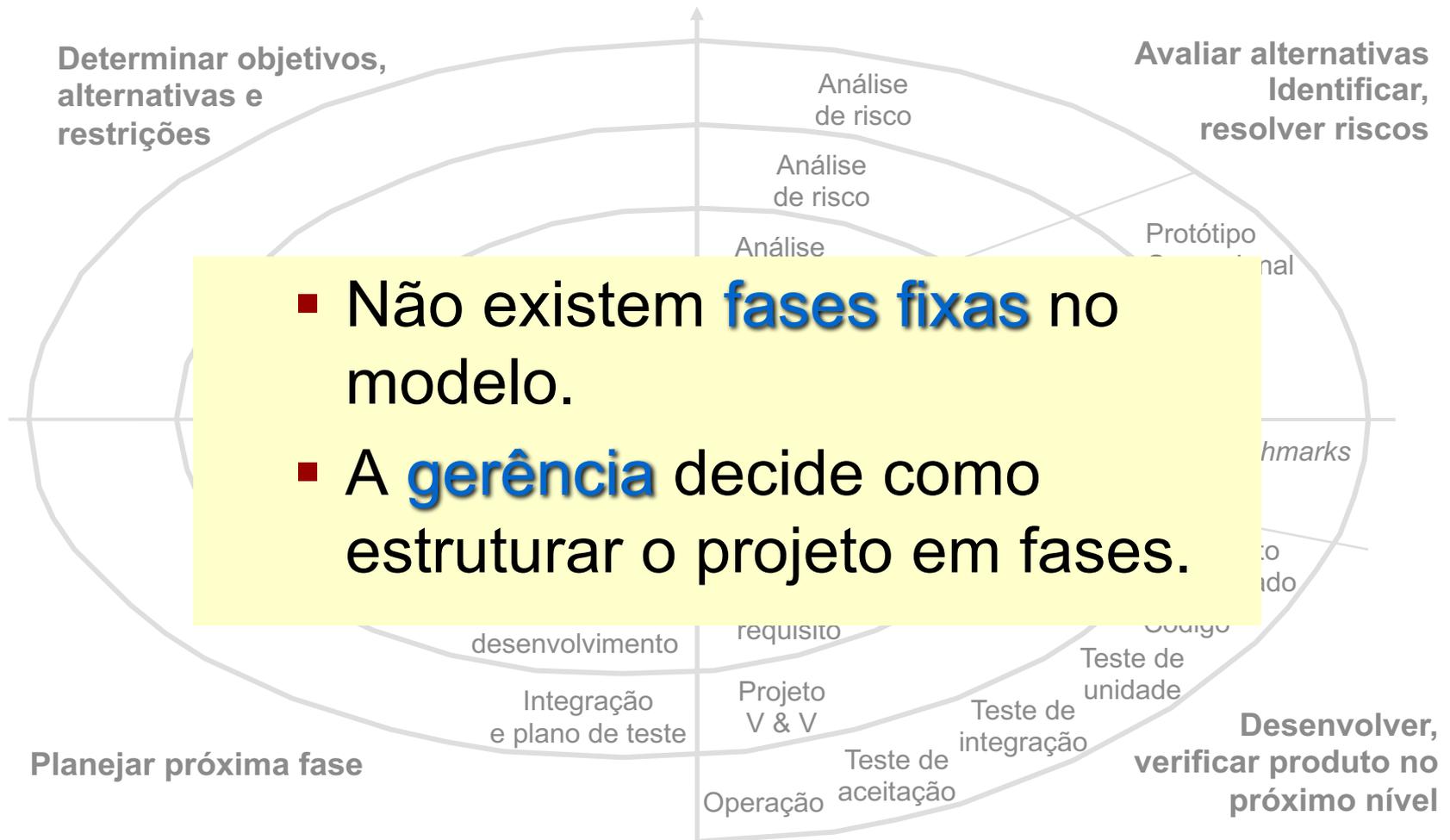
O Modelo Espiral

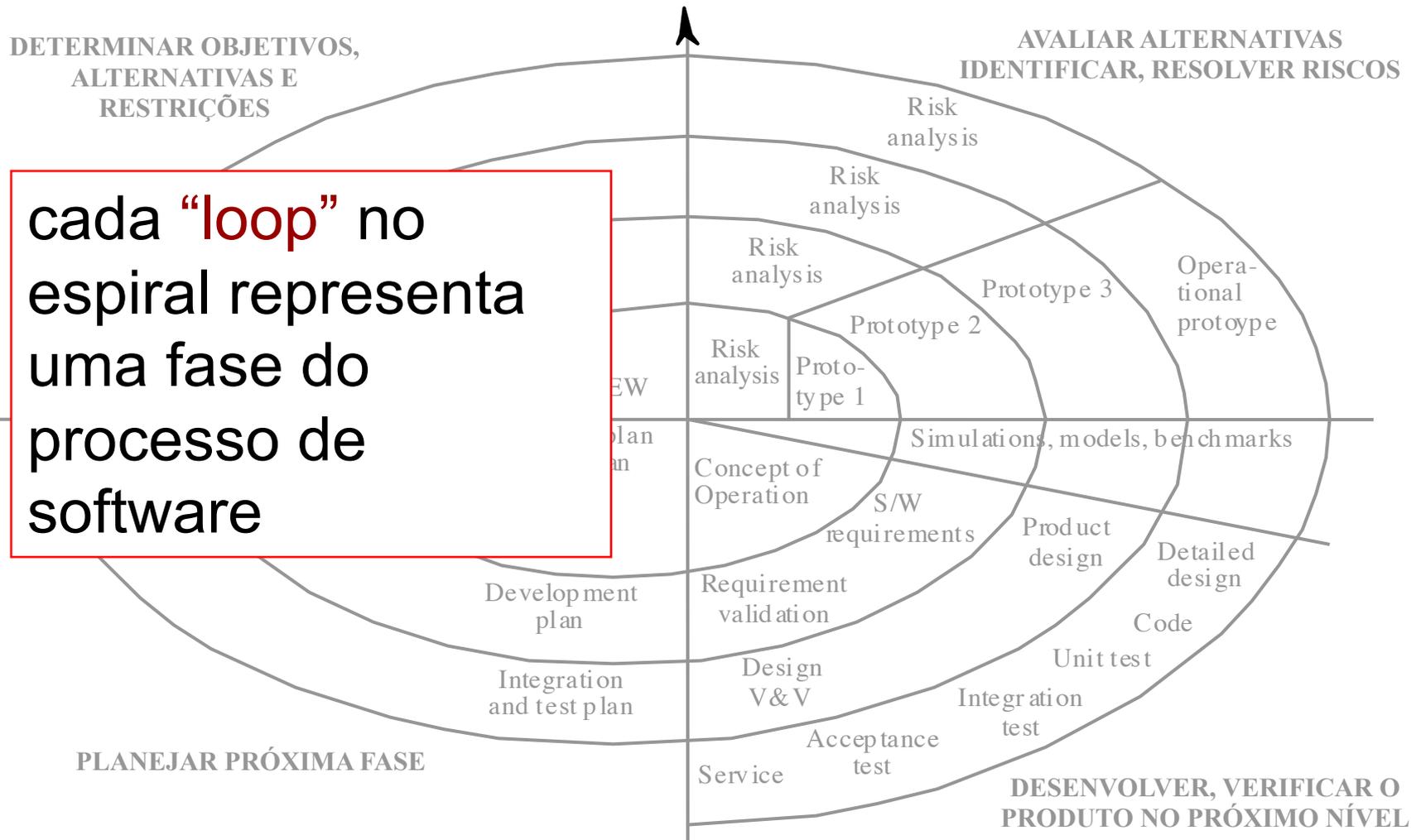


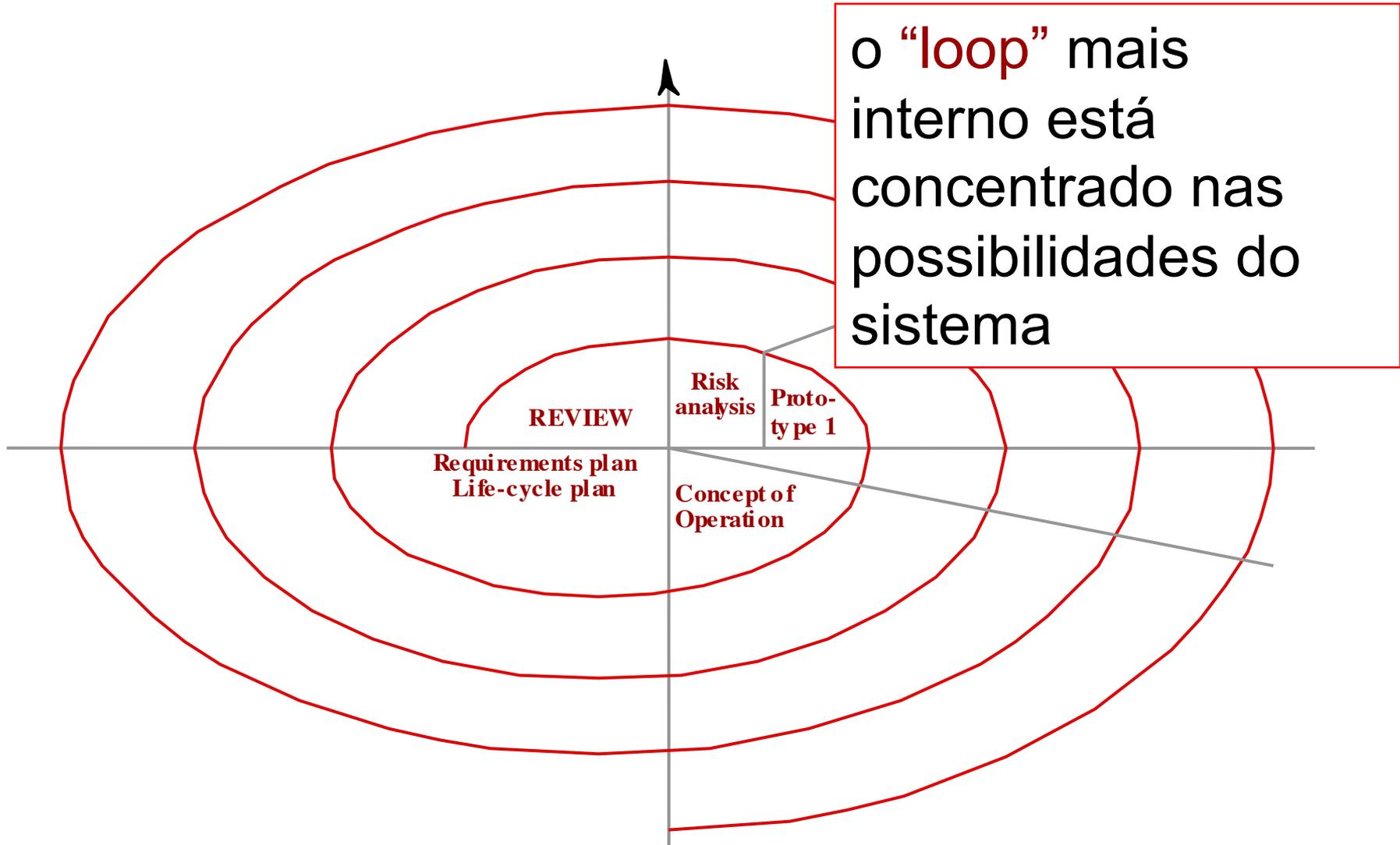
O Modelo Espiral



O Modelo Espiral







o próximo “loop”
está concentrado na
definição dos
requisitos do
sistema

Risk
analysis

Prototype

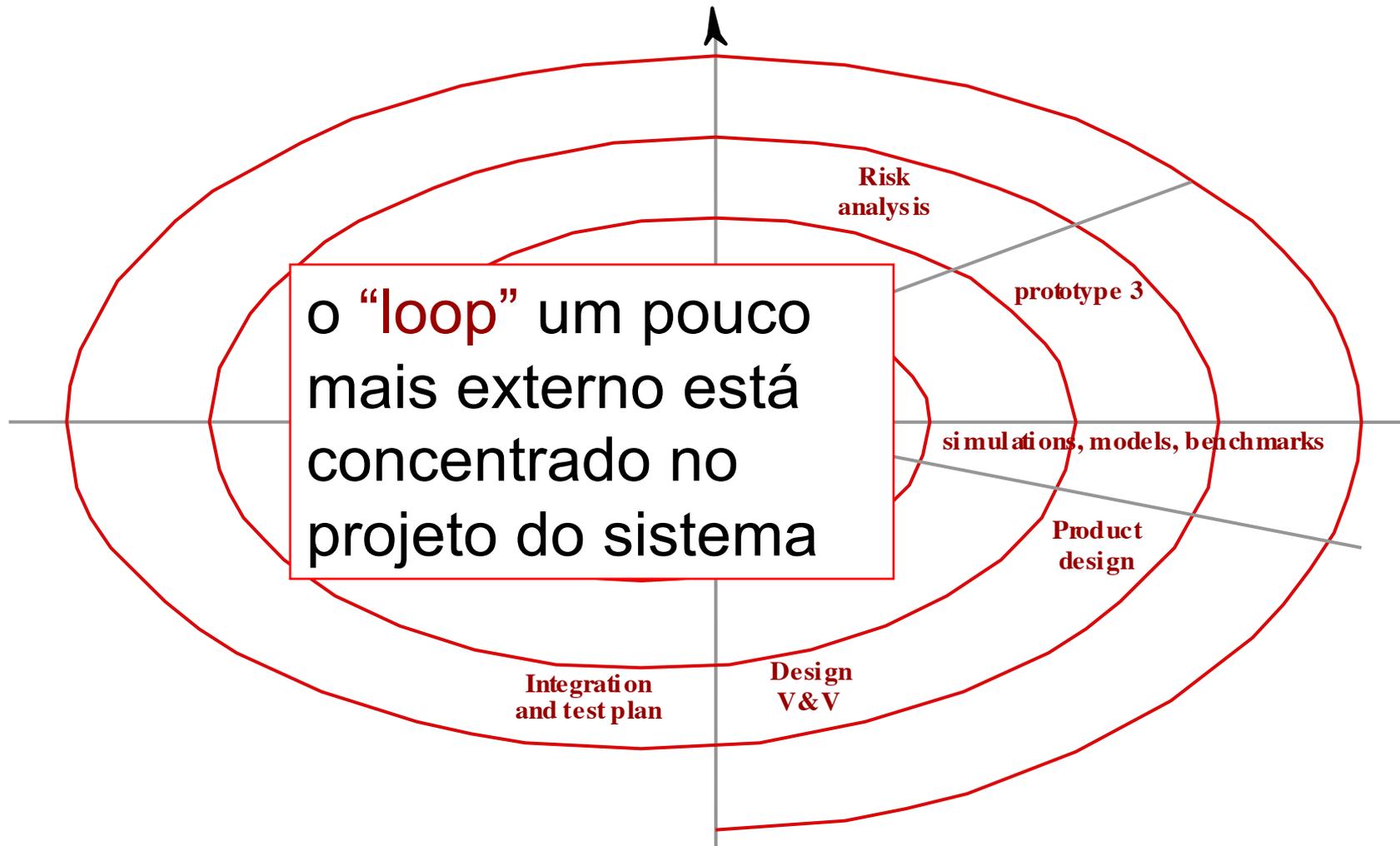
Simulations, models, benchmarks

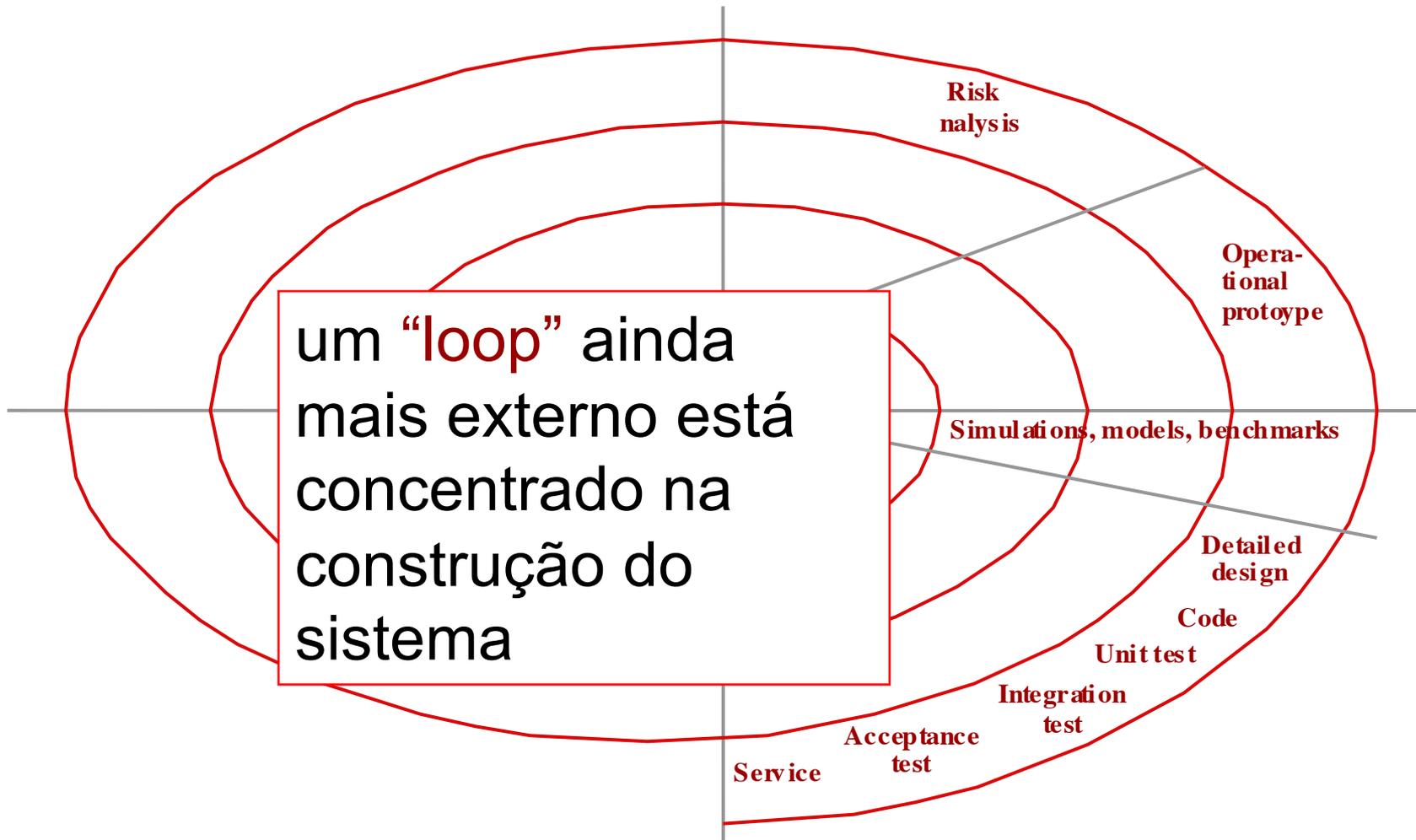
SW
requirements

Development
plan

Requirement
validation







Comentários sobre o Ciclo de Vida em Espiral

- Se o projeto não puder ser concluído por razões técnicas, isso é descoberto cedo, antes de um grande investimento ser feito
 - As primeiras voltas da espiral são mais baratas
- Porém, exige gerência complexa e eficiente

Comentários sobre o Ciclo de Vida em Espiral

- pode ser difícil convencer os clientes que uma abordagem "evolutiva" é controlável
 - exige considerável experiência na determinação de riscos e depende dessa experiência para ter sucesso
- o modelo é relativamente novo e não tem sido amplamente usado. Demorará muitos anos até que a eficácia desse modelo possa ser determinada com certeza absoluta

Modelos de Processo de Software

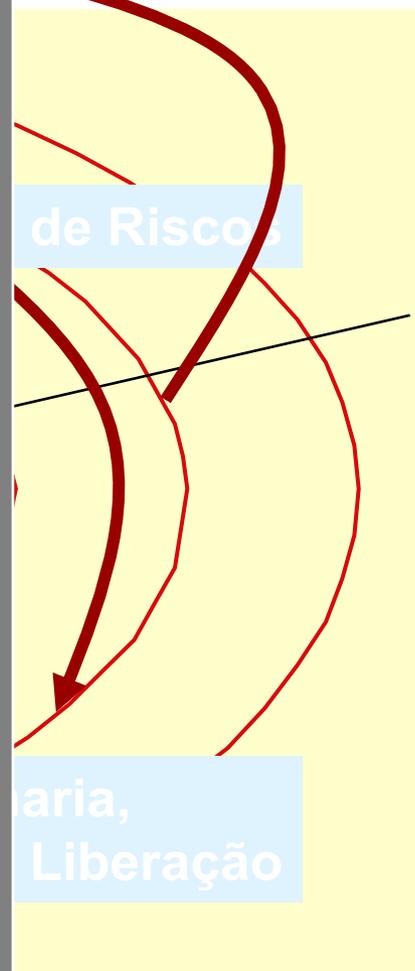
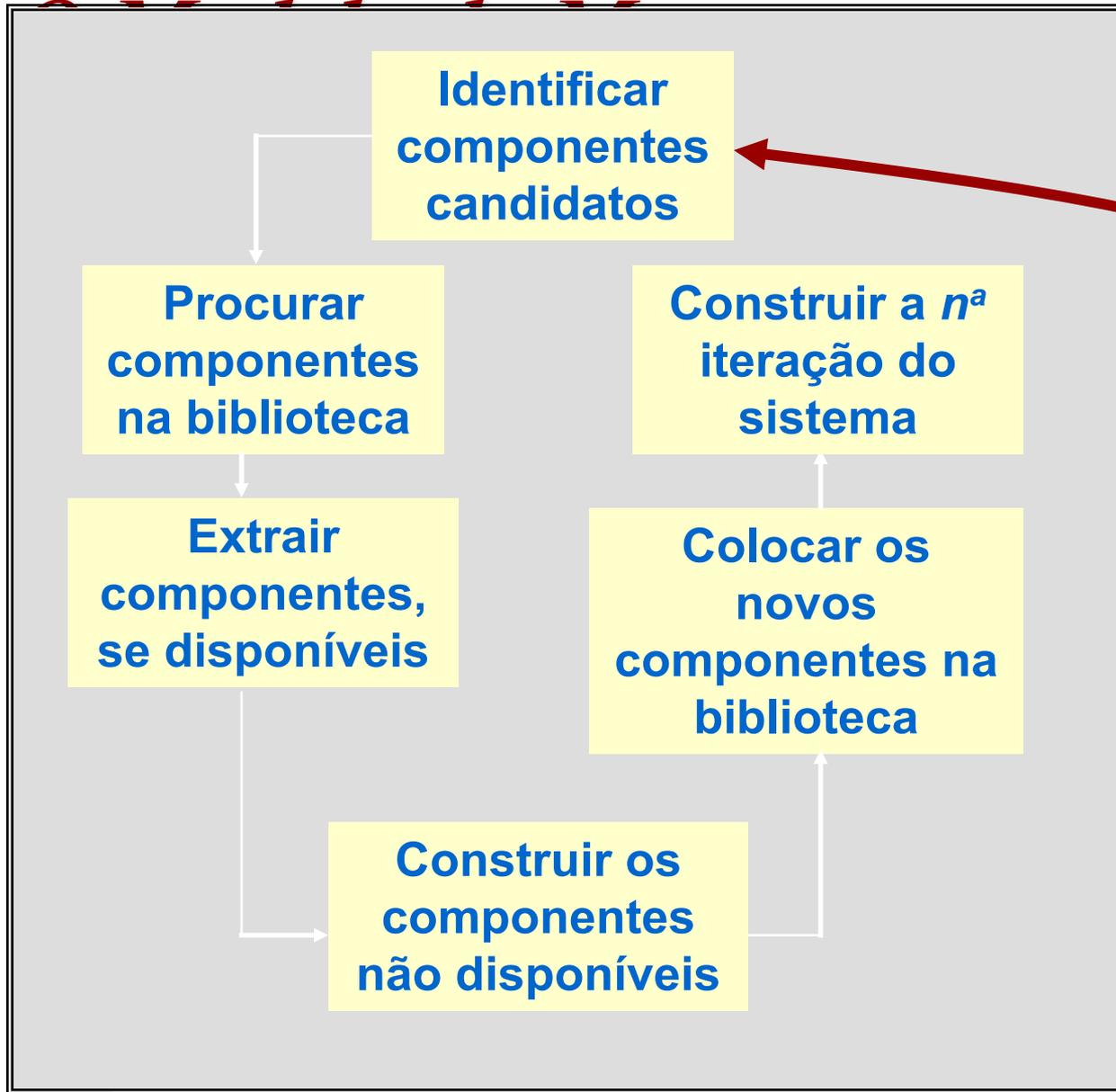
- *O Modelo Sequencial Linear*
 - *também chamado Modelo Cascata*
- *O Modelo de Prototipação*
- *O Modelo RAD (Rapid Application Development)*
- **Modelos Evolutivos de Processo de Software**
 - *O Modelo Incremental*
 - *O Modelo Ágil*
 - *O Processo Unificado*
 - *O Modelo Espiral*
 - *O Modelo de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

O Modelo de Montagem de Componentes

- Incorpora características de **tecnologias orientadas a objeto** no **Modelo Espiral**.
- É de natureza **evolutiva** demandando uma abordagem **iterativa** para a criação do software.
- O modelo **compõe** aplicações a partir de componentes de software “**empacotados**”.
 - Quando projetadas e implementadas apropriadamente, as classes são **reutilizáveis** em diferentes aplicações e arquiteturas de sistema.

O Modelo de Montagem de Componentes





O Modelo de Montagem de Componentes

- O modelo de montagem de componentes conduz ao **reúso** do software.
- A reusabilidade pode fornecer uma série de benefícios:
 - Redução no tempo de desenvolvimento.
 - Redução no custo do projeto.
 - Aumento no índice de produtividade.
 - Tais benefícios dependem da **robustez da biblioteca** de componentes.

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 Ágil*
 - *O Processo Unificado*
 - *O Modelo Espiral*
 - *O Modelo de Montagem de Componentes*
- ***Modelo de Métodos Formais***
- ***Técnicas de Quarta Geração***

O Modelo de Métodos Formais

- Especificar, desenvolver e verificar um sistema pela aplicação de uma **rigorosa notação matemática**.
 - Abrange um conjunto de atividades que levam à **especificação matemática formal** do software.
 - Tem como base a **transformação matemática formal** de uma especificação de sistema em um programa executável.

O Modelo de Métodos Formais

LearningOrg

FormalOrg

learningteams : \mathbb{P} *LearningTeam*

sharedvisions : \mathbb{P} *SharedVision*

sharedvisions \neq

$\forall lt : LearningTeam \mid lt \in learningteams \bullet$

$(\exists_1 te : Team \mid te \in teams \bullet$

$te = (\lambda LearningTeam \bullet \theta Team) lt)$

$\#teams = \#learningteams$

O Modelo de Métodos Formais

- No projeto...
 - Base para a **verificação do programa**, permitindo que se descubram erros que poderiam passar despercebidos.
- No desenvolvimento...
 - **Ambigüidade**, **não completitude** e **inconsistência** podem ser mais facilmente descobertas pela análise matemática.

O Modelo de Métodos Formais

- Abordagem particularmente adequada ao desenvolvimento de sistemas com rigorosas exigências de **segurança**, **confiabilidade** e **garantia**.
 - Eletrônica de aeronaves.
 - Controle de tráfego aéreo.
 - Dispositivos médicos.
 - Telecomunicações.

Modelo de Métodos Formais: Problemas

- Preocupações quanto à aplicabilidade em ambientes comerciais:
 - O desenvolvimento é **lento** e **dispendioso**.
 - Requer **treinamento extensivo**.
 - Difícil utilizar os modelos como um mecanismo de **comunicação**.
 - Clientes despreparados tecnicamente.

Cleanroom (Sala Limpa)

- **Exemplo** mais conhecido do processo de desenvolvimento formal.
 - Desenvolvimento **incremental** do software.
 - Cada estágio é desenvolvido e sua correção é demonstrada utilizando-se uma **abordagem formal**.
 - O teste de sistema visa a medir a **confiabilidade** do sistema.
 - Conduzido como um experimento estatístico.

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 Ágil*
 - *O Processo Unificado*
 - *O Modelo Espiral*
 - *O Modelo de Montagem de Componentes*
- *Modelo de Métodos Formais*
- ***Técnicas de Quarta Geração***

Técnicas de Quarta Geração

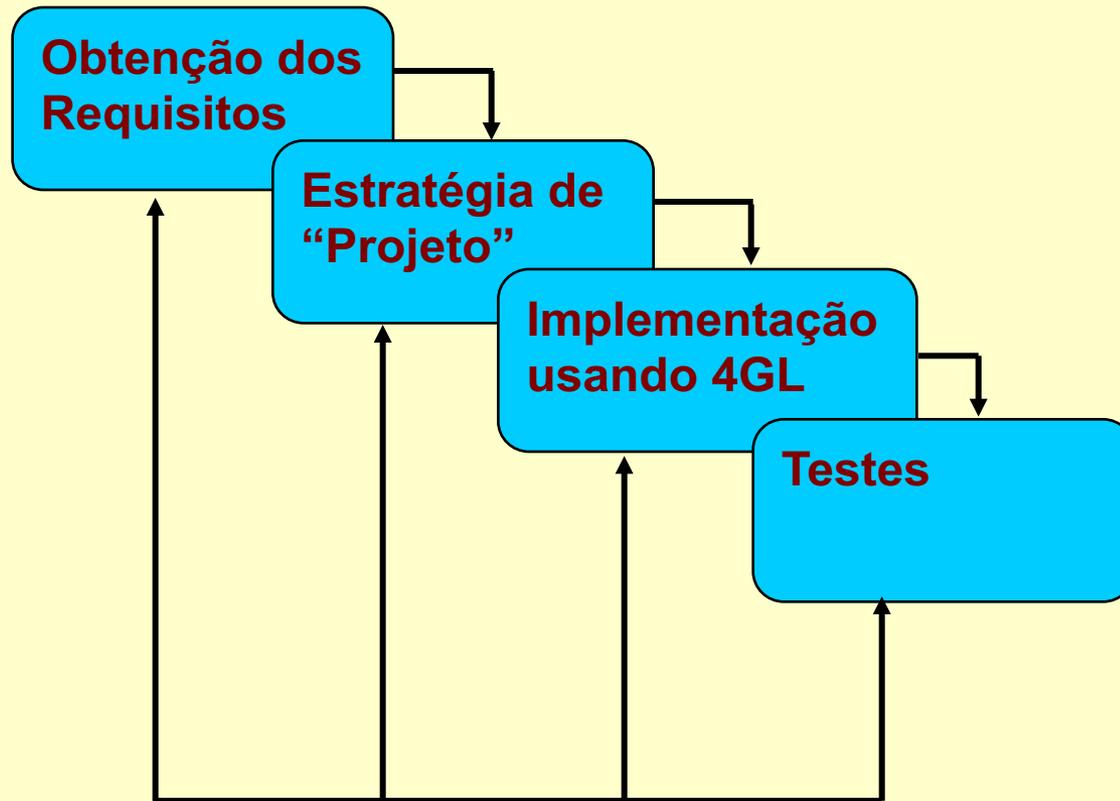
- Engloba um conjunto de ferramentas de software possibilitando que:
 - O sistema seja especificado em uma **linguagem de alto nível**.
 - O código-fonte seja gerado **automaticamente** a partir dessas especificações.

Técnicas de Quarta Geração

➤ Ferramentas de Apoio

- Linguagens não-procedimentais para:
 - Consulta de banco de dados (*SQL*).
 - Geração de relatórios (*PostScript*).
 - Manipulação de dados (*XML*).
 - Interação e definição de telas (*Delphi, Visual Basic*).
 - Geração de código.
- Capacidade gráfica de alto nível.
- Capacidade de disposição em planilhas.
- Geração automática de HTML e linguagens similares para criação de páginas Web.

Técnicas de Quarta Geração



Técnicas de Quarta Geração

Obtenção dos Requisitos

- O cliente descreve os requisitos os quais são traduzidos para um protótipo operacional.
 - O cliente pode estar inseguro quanto aos requisitos.
 - O cliente pode ser incapaz de especificar as informações de um modo que uma ferramenta 4GL possa ser utilizada.
 - As 4GLs atuais não são suficientemente sofisticadas para acomodar a verdadeira linguagem natural.

Técnicas de Quarta Geração

Estratégia de “Projeto”

- Pequenas Aplicações
 - É possível mover-se do passo de Obtenção dos Requisitos para o passo de Implementação usando uma linguagem de quarta geração.
- Grandes Projetos
 - É necessário desenvolver uma estratégia de projeto.
 - De outro modo ocorrerão os mesmos problemas encontrados quando se usa abordagem convencional (baixa qualidade).

Técnicas de Quarta Geração

Implementação usando 4GL

- Os resultados desejados são representados de modo que haja geração automática de código.
- Deve existir uma estrutura de dados com informações relevantes e que seja acessível pela 4GL.

Técnicas de Quarta Geração

Testes

- O desenvolvedor deve efetuar testes e desenvolver uma documentação significativa.
- O software desenvolvido deve ser construído de maneira que a manutenção possa ser efetuada prontamente.

Técnicas de Quarta Geração

➤ Prós

- Redução dramática no **tempo** de desenvolvimento do software.
 - Aumento de **produtividade**.

➤ Contras

- As 4GL atuais não são mais fáceis de usar do que as linguagens de programação.
- O código-fonte produzido é ineficiente.
- A **manutenibilidade** de sistemas usando técnicas 4GT ainda é questionável.

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 Ágil*
 - *O Processo Unificado*
 - *O Modelo Espiral*
 - *O Modelo de Montagem de Componentes*
- *Modelo de Métodos Formais*
- *Técnicas de Quarta Geração*

Concluindo...

- *Cascata, Prototipação, Evolutivos, Formais, Ágeis, ...*
- Para escolha de um **modelo de processo de software**:
 - Natureza do projeto e da aplicação.
 - Métodos e ferramentas a serem usados.
 - Controles e produtos que precisam ser entregues.
- Não são mutuamente exclusivos e freqüentemente são usados em conjunto.