

Modelos de desenvolvimento de software

Paradigmas de engenharia de software

São modelos de processo de desenvolvimento de sistemas que especificam quais atividades devem ser executadas e em qual ordem (oposta a uma "abordagem casual")!

Ciclo de vida clássico(cascata)	Evolutivo (dep. Prototipação)	Espiral
<p>Dividido em etapas ...</p> <p>* Eng. sistemas</p> <ul style="list-style-type: none"> * Análise de requisitos * Planejamento de Projeto * Codificação * Testes * Manutenção <p>• Método sequencial - o resultado de uma fase é a entrada da outra.</p> <p>• Indicado quando os requisitos estão bem claros.</p> <p>• Prima por um fluxo sequencial de atividades na tentativa de manter o processo previsível e fácil de monitorar.</p> <p>• Rigidez: não prevê dinamismo nos requisitos ou interações entre as etapas, não acomoda incertezas típicas do começo de muitos projetos.</p> <p>• Exige paciência do cliente, o desenvolvimento pode ser muito longo.</p>	<p>Um produto inicial é desenvolvido e implementado e vai sendo refinado através de múltiplas versões até que o sistema almejado tenha sido obtido. Assim este modelo atende bem os requisitos do cliente (gera produto: desenvolvimento exploratório) - é um bom mecanismo para identificar requisitos (não gera produto, serve só para esclarecer requisitos - protótipo descartável).O processo não é visível, não compensa documentar cada versão da implementação. Sistema é pobremente estruturado dadas as mudanças constantes. Distância entre protótipo e produto ("rapidez") representa um risco que sacrifica a qualidade - Melhor para sist. pequenos !</p>	<p>Combina o ciclo de vida clássico com o evolutivo adicionando Análise de Risco.</p> <p>O desenvolvimento se organiza como uma espiral que tem muitos ciclos - cada ciclo é uma fase do processo de desenvolvimento.</p> <p>Parte do princípio de que a forma do desenvolvimento não pode ser completamente determinada de antemão.</p> <p>Desvantagem: custo.</p>

Ciclo de vida clássico ou modelo em cascata

Observação importante: cada uma das atividades fornecem um feed-back para as fases anteriores, tal feed-back estimula a melhoria e a evolução do software.

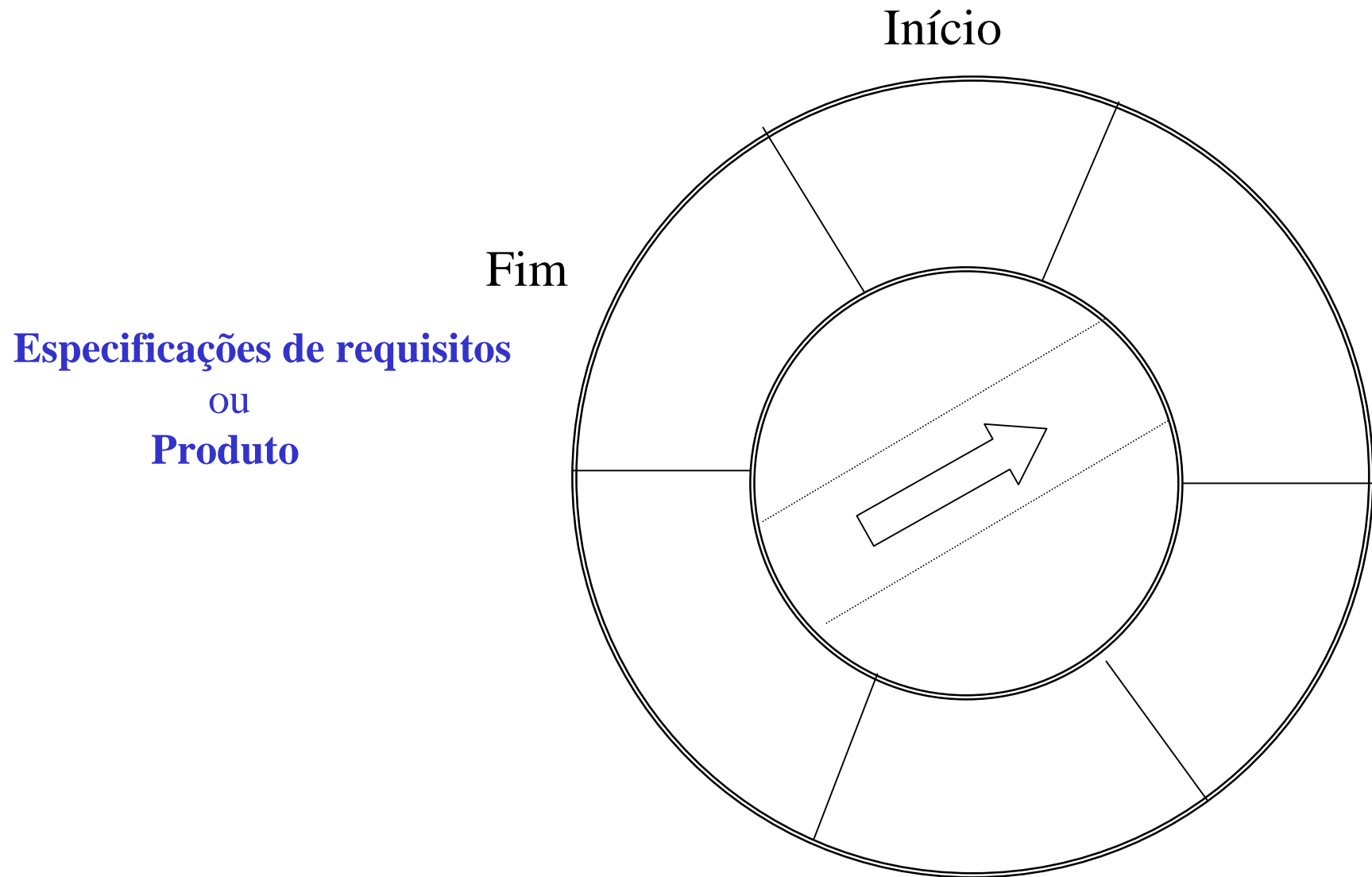
O quê

Como

Operação

Eng. Sistemas	Análise de requisitos	Planejamento de Projeto	Codificação	Testes	Manutenção
O software sempre faz parte de um sistema mais amplo - deve fazer interface com outros elementos tais como hardware, pessoas e bancos de dados.	Identificação das restrições e dos serviços e metas a serem atingidas (determinar <i>quais</i> e não <i>como..</i>). Esta fase gera um documento de especificação de requisitos para (a) o cliente verificar se satisfaz suas expectativas (b) ser usado pelos desenvolvedores de software.	Processo de múltiplos passos: estrutura de dados, arquitetura de software, interface. Como os requisitos, o projeto é documentado e torna-se parte da configuração do software: documento de especificação do projeto.	Fase em que o projeto de software se transforma em um conjunto de programas.	Testes tanto nos aspectos lógicos internos do programa, quanto nos aspectos externos (entradas e saídas). Testes são feitos progressivamente, em conjuntos cada vez maiores, a partir de pequenos subsistemas até que o sistema inteiro seja construído.	O software poderá sofrer mudanças depois de entregue: acomodar mudanças, erros encontrados, acréscimos etc. A manutenção replica cada uma das etapas precedentes. Pode vir a ser a etapa mais longa do ciclo de vida.

Modelo Evolutivo



Modelo Evolutivo

- Prototipação é um método rápido de desenvolvimento.
- No passado desenvolver por protipação era considerado um método inferior que exigiria grandes esforços adicionais.. porém muitos sistemas hoje são desenvolvidos pelo método evolucionário, ao menos parcialmente.
- Prototipação como uma atividade de redução de riscos.
- O uso principal é ajudar clientes e desenvolvedores a compreender requisitos do sistema:
 - Usuários podem experimentar com um protótipo para verificar o comportamento do sistema.
 - O protótipo pode revelar erros e omissões nos requisitos (validação).

Vantagens do modelo Evolutivo

- Diferenças entre percepções do cliente e dos desenvolvedores são explicitadas.
- Um sistema funcional é apresentado antecipadamente.
- O protótipo pode servir de base para a especificação do sistema.
- O protótipo pode entrever necessidade de treinamento de pessoas e esquema de testes.
- Aumento da usabilidade, manutenibilidade e até mesmo da qualidade do sistema, se tomado para amadurecer o entendimento dos requisitos.
- Prototipação é interessante para desenvolver partes do sistema de difícil especificação como interface do usuário.
- Cliente envolve-se na avaliação do protótipo.

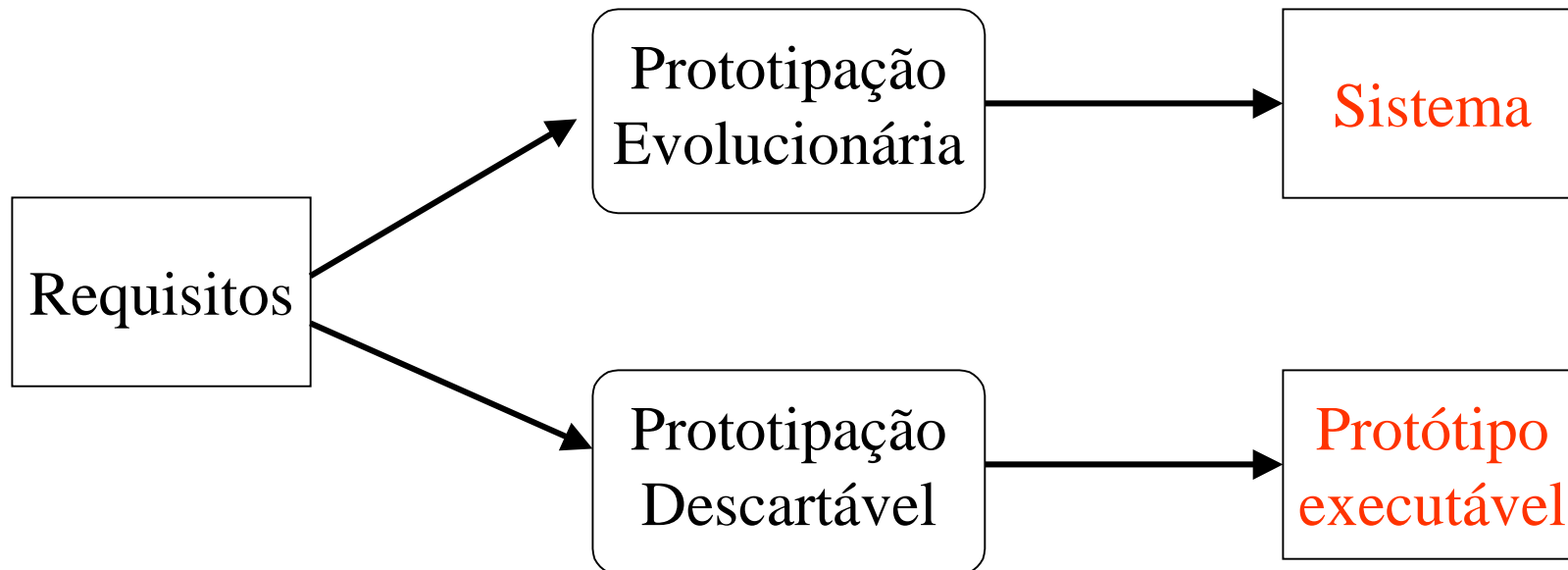
Desvantagens do modelo Evolutivo

- Questões de gerenciamento
 - Esquema muito menos estruturado
 - Specialist skills are required which may not be available in all development teams
- Problemas de manutenção
 - Constantes mudanças tendem a corromper a estrutura do sistema o que pode repercutir na necessidade de manutenção (ou retrabalho) mais cara.
- Problemas contratuais (entrega antes do prazo de um sistema mas com qualidade inferior).

Duas variantes de prototipação:

- **Prototipação evolucionária**
 - Um protótipo inicial é produzido e refinado até que se tenha um produto final para os usuários finais.
 - Parte-se dos requisitos melhor compreendidos.
- **Prototipação descartável**
 - Um protótipo, usualmente uma implementação do sistema, é produzido para ajudar a compreender os requisitos do sistema e então é descartado. O sistema é então desenvolvido usando-se outro processo de desenvolvimento.
 - Parte-se dos requisitos pobremente compreendidos.

Modelos de prototipação:

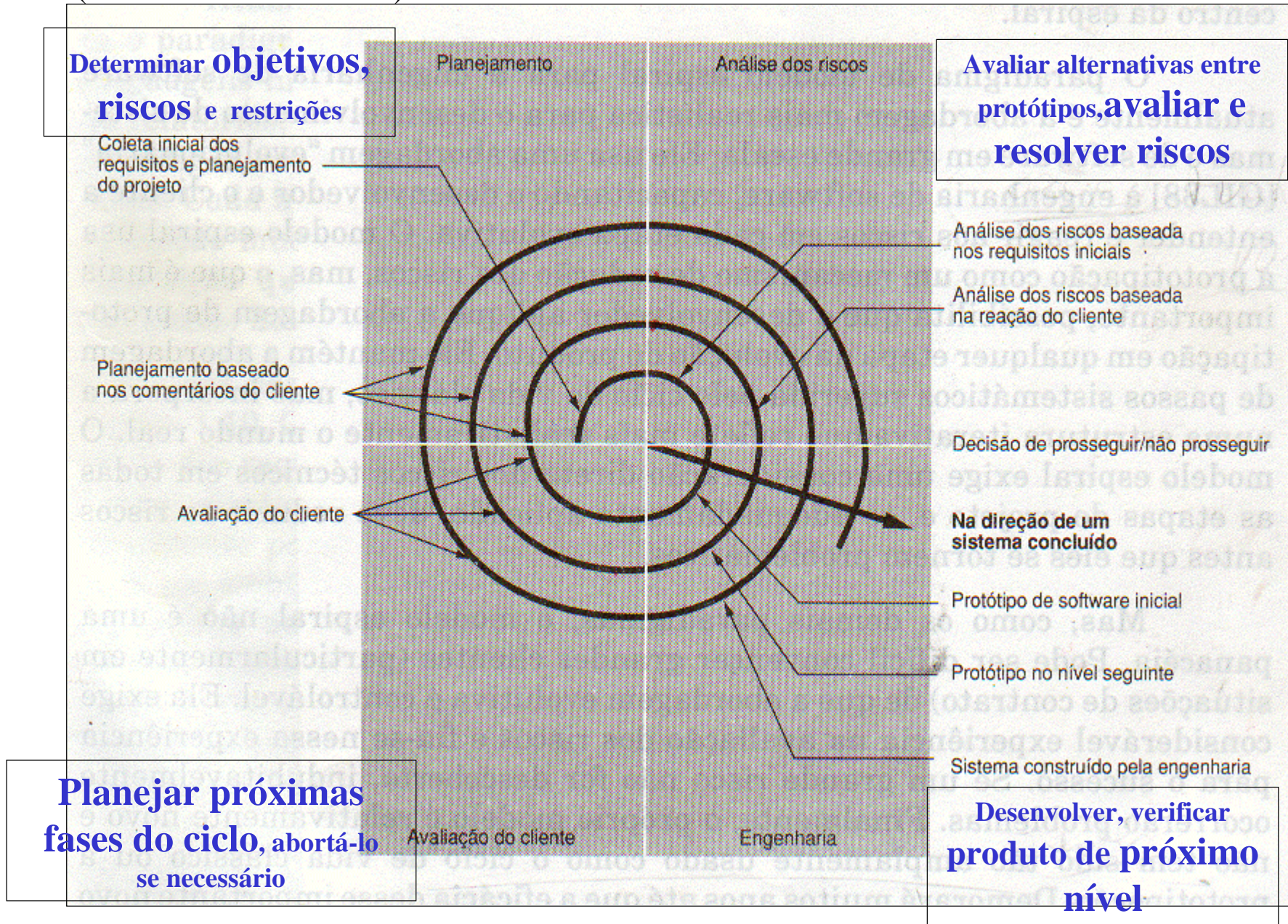


Modelo Espiral

- Processo é representado por uma espiral em lugar de uma sequência linear.
- Cada loop na espiral representa uma fase do processo: loop mais interno pode representar a viabilidade; o seguinte o planejamento e assim por diante.
- Riscos são regularmente avaliados durante o processo.
- Um ciclo da espiral começa com a elaboração de objetivos, como desempenho, funcionalidade, etc. Meios alternativos de atingir esses objetivos e suas restrições são enumerados. Cada alternativa é examinada em relação a cada objetivo. Isso resulta na identificação das causas de riscos. A próxima etapa é avaliar esses riscos. Em seguida uma parte é realizada e parte-se para a próxima fase do projeto.

Modelo Espiral (Boehm)

(Versão Pressman/ Peters)

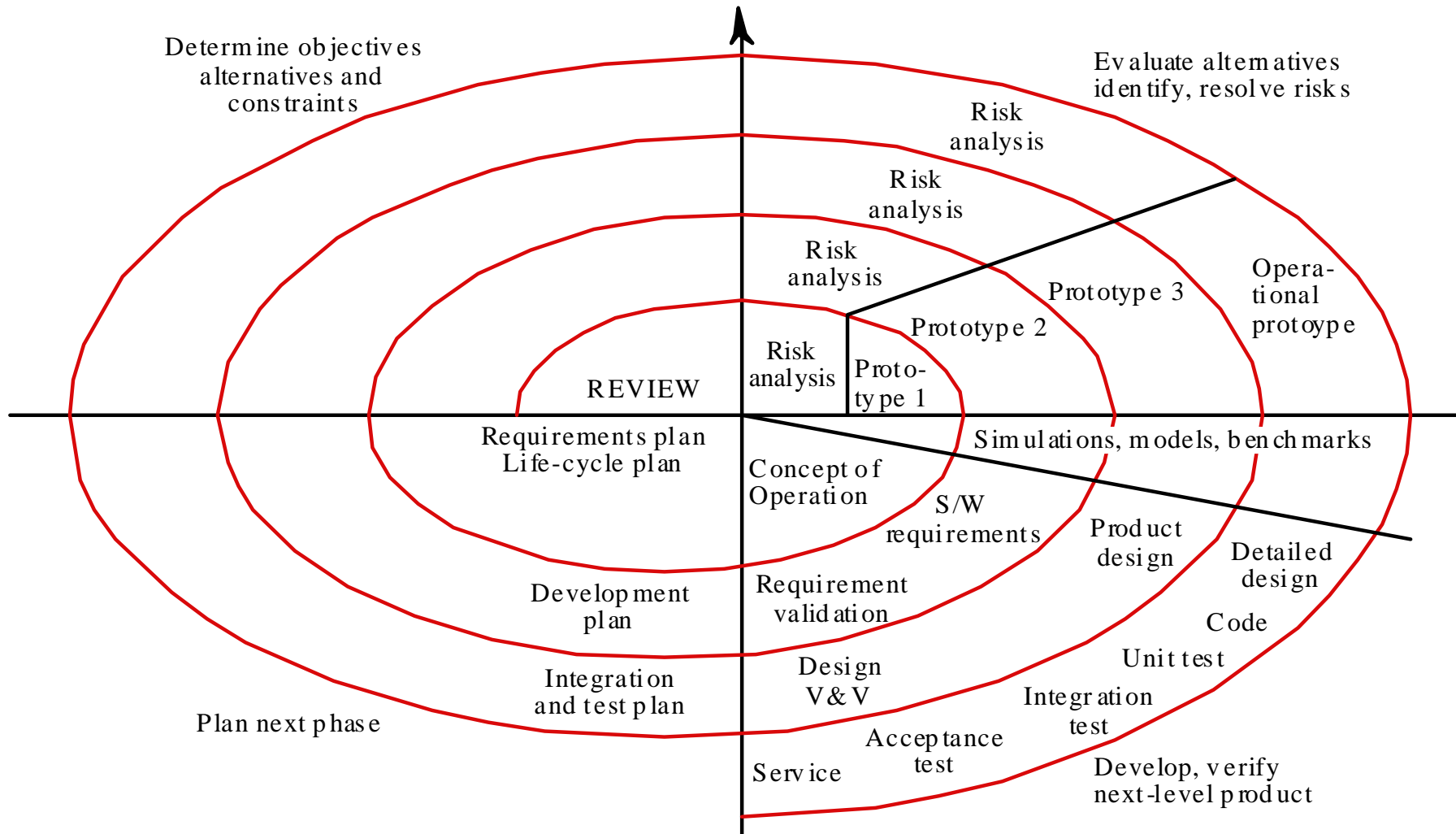


Setores do modelo Espiral

- **Determinação de objetivos**
 - São identificados objetivos específicos
- **Análise e redução de riscos**
 - Riscos são detectados e atividades são acionadas para reduzir riscos
- **Desenvolvimento e Validação**
 - Um modelo de desenvolvimento é escolhido
- **Planejamento**
 - O projeto é revisado e a próxima fase da espiral é planejada

Modelo Espiral (Boehm)

(Versão Sommerville)



Observações

- Os pontos fortes de cada um dos paradigmas podem ser utilizados em um mesmo projeto.
- O paradigma espiral já faz isso diretamente, combinando prototipagem e elementos do ciclo de vida clássico em uma paradigma evolutivo. Qualquer um pode servir como alicerce no qual os outros paradigmas podem ser integrados. O processo sempre começa com a determinação dos objetivos, alternativas e restrições (requisitos). Se os requisitos não estiverem muito claros, um protótipo pode ser usado para melhor defini-lo. Usando o protótipo como guia, o desenvolvedor pode retornar aos passos do ciclo de vida clássico. Alternativamente o protótipo pode evoluir para um sistema.
- A natureza da aplicação é que vai determinar o paradigma a ser utilizado e a combinação de paradigmas só tende a beneficiar o processo como um todo.