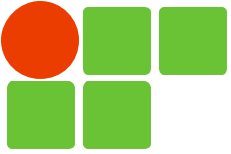


INSTITUTO FEDERAL
ESPIRITO SANTO

CAMPUS COLATINA

Modelos de Processo de Software

Prof. Victorio Albani de Carvalho

Introdução

Há décadas atrás:

- Softwares eram construídos para áreas específicas;
- Para poucos usuários utilizarem;
- Prazos não eram críticos;
- Produção “artesanal” de software, por equipes pequenas ou até mesmo individuais.

Introdução

Com a disseminação dos computadores

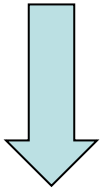
- Produz-se software para milhares de usuários;
- Há uma necessidade crescente de integração e de manutenção;
- Prazos cada vez mais críticos;
- Equipes envolvidas cada vez maiores;
- Variedade de tecnologias;

Introdução

- Nesse novo contexto desenvolver Software não é simplesmente uma atividade de programação!
- É preciso sistematizar o desenvolvimento de software!
- Precisamos da engenharia de software!

Engenharia de Software

O objetivo da Engenharia de Software é construir, de maneira eficiente, software de qualidade



Para tanto, ela procura evoluir e aplicar, de forma adequada, métodos, técnicas e ferramentas

Processo de Software

Para que os métodos e ferramentas sejam aplicadas de forma adequada e sejam passíveis de controle, estes são relacionados no contexto de um

PROCESSO

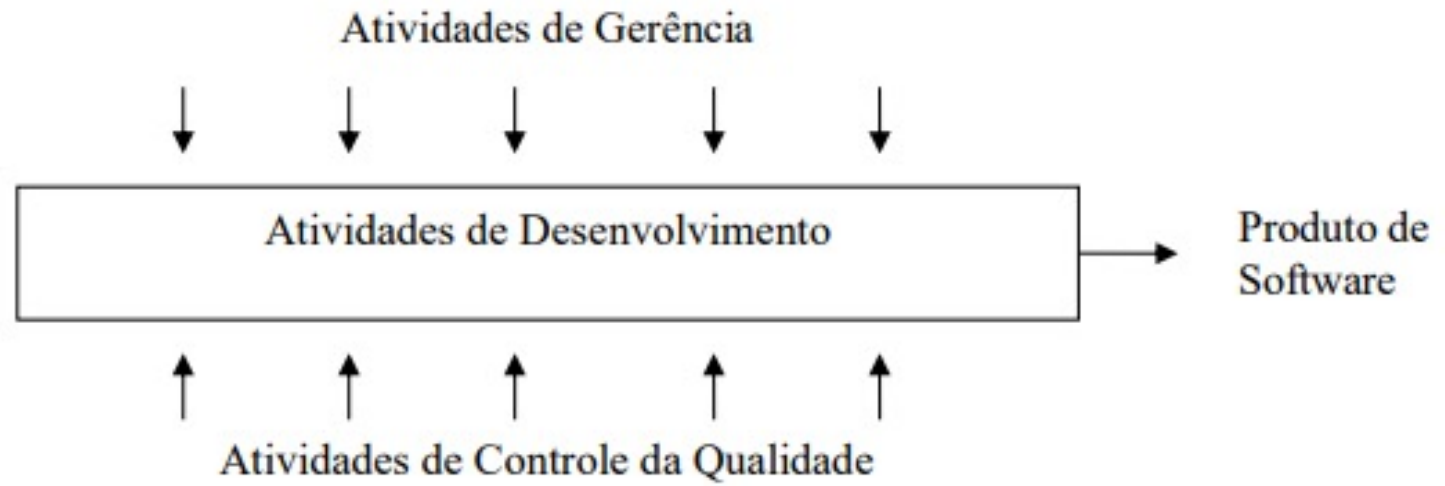
Processo de Software

Um processo de software pode ser visto como o conjunto de atividades, métodos, práticas e transformações que guiam pessoas na produção de software

Tipos de Atividades do Processo de Software

- As atividades de um processo de software podem ser classificadas quanto ao seu propósito em:
  - **Atividades de Desenvolvimento**: Contribuem diretamente para o desenvolvimento do produto, como especificação de requisitos, análise, projeto e implementação.
  - **Atividades de Gerência**: são relacionadas ao planejamento como realização de estimativas, elaboração de cronogramas e análise de riscos
  - **Atividades de Garantia da Qualidade**: visam a garantia da qualidade do produto e do processo, tais quais revisões e inspeções de produtos.

Tipos de Atividades do Processo de Software



As **atividades de desenvolvimento** formam a espinha dorsal do desenvolvimento e são realizadas segundo uma ordem estabelecida no planejamento.

As **atividades de gerência e de controle da qualidade** são ditas atividades de apoio, pois não estão ligadas diretamente à construção do produto final (software e documentação). Essas atividades, normalmente, são realizadas ao longo de todo o ciclo de vida.

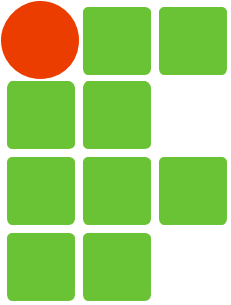
Atividades Habituais do Ciclo de Vida do Software

- Planejamento
- Levantamento de Requisitos
- Análise e Especificação de Requisitos
- Projeto
- Implementação
- Testes
- Entrega e Implantação
- Operação
- Manutenção

Modelos de ciclo de vida

Mas como as atividades são encadeadas dentro do processo???

Um modelo de ciclo de vida (ou modelo de processo) é uma representação abstrata de um esqueleto de processo



INSTITUTO FEDERAL
ESPIRITO SANTO
CAMPUS COLATINA

Modelos de Ciclo de Vida

Modelos Sequenciais

1. Modelo em Cascata
2. Modelo em V

Modelos Incrementais

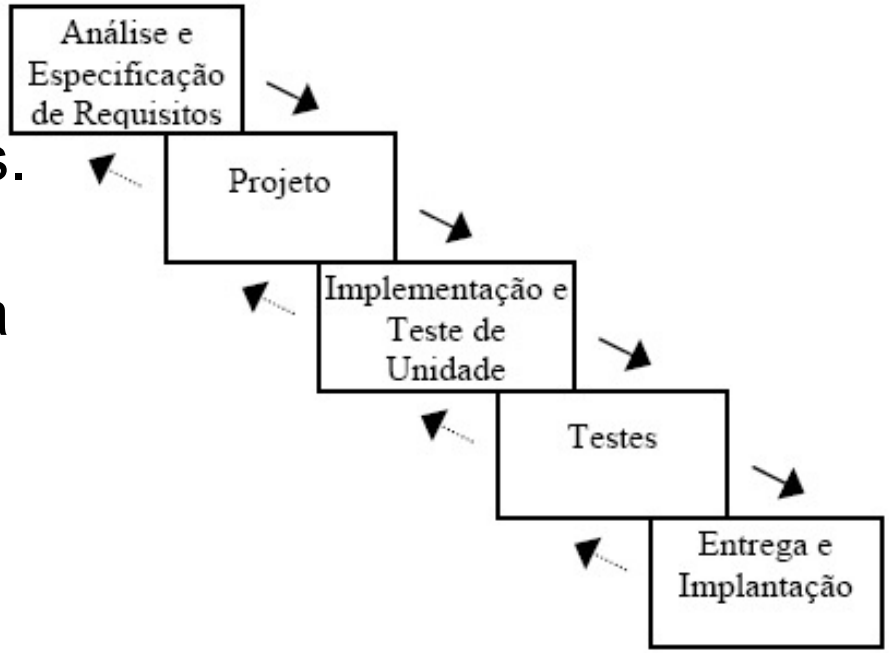
1. Modelo Incremental
2. Modelo RAD (Rapid Application Development)

Modelos Evolutivos

1. Modelo em Espiral
2. Modelo do Processo Unificado (RUP)

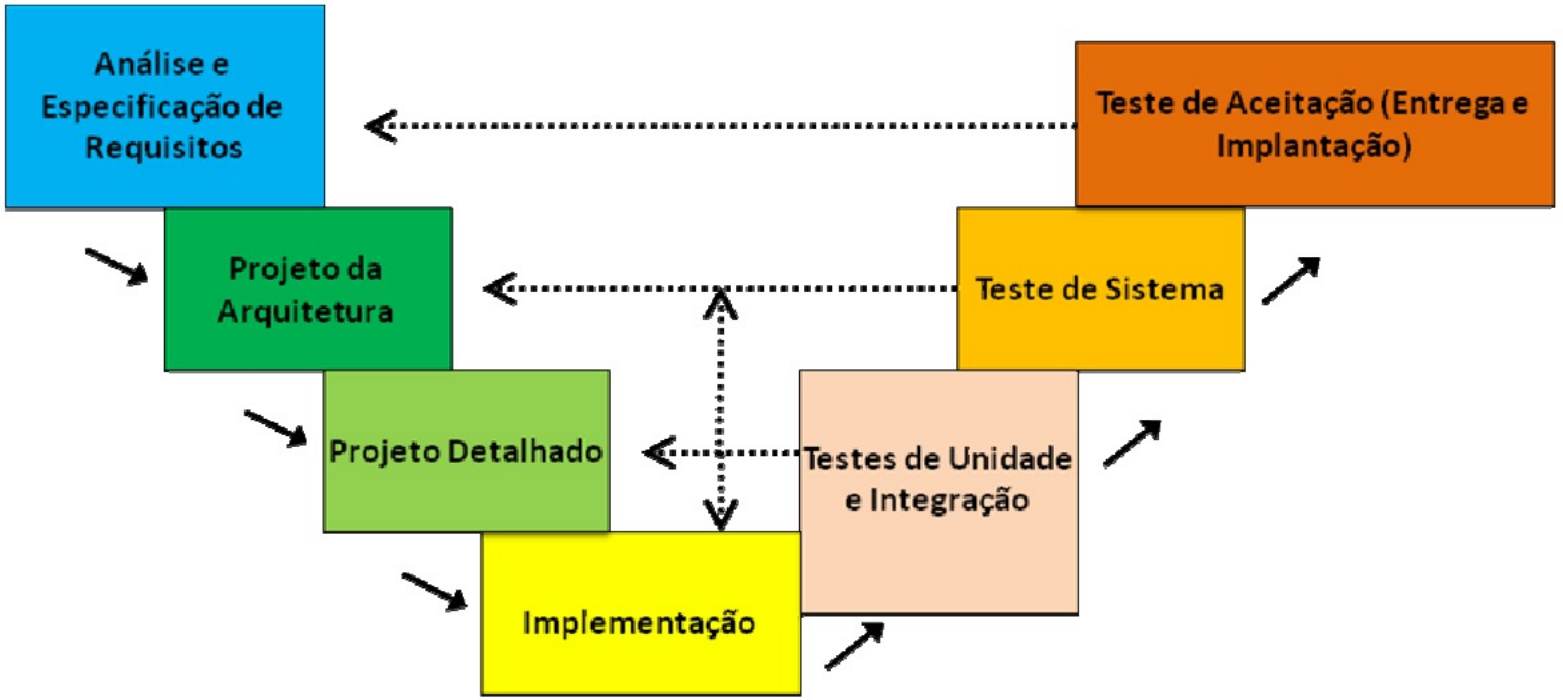
Modelo em Cascata

- É o modelo mais antigo.
- As atividades são executadas de forma sequencial. Uma só inicia após a conclusão da fase precedente.
- Revisões ao fim de cada fase com o envolvimento do usuário.
- É permitido retorno à fase anterior para correção de erros.
- Cada fase serve como uma base aprovada e documentada para o passo seguinte, facilitando bastante a gerência de configuração.



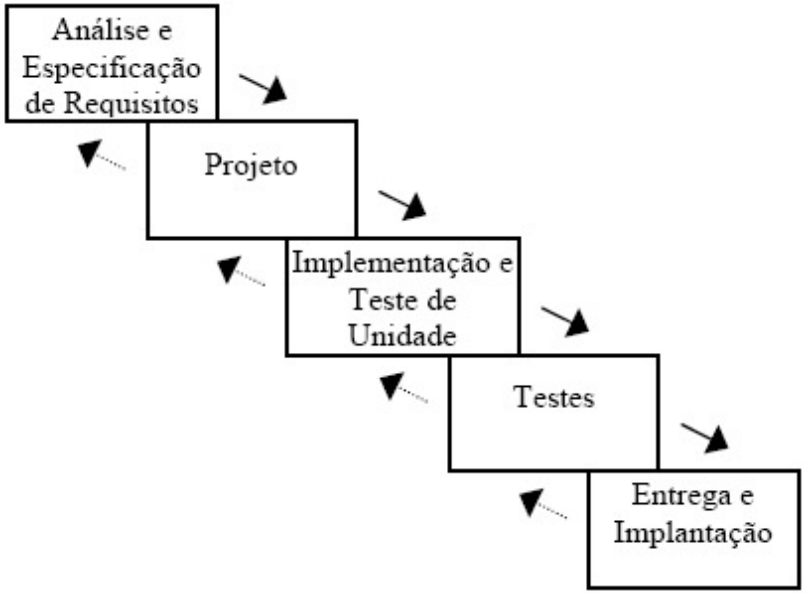
Modelo em V

- Variação do modelo em cascata.
- Enfatiza a relação entre as atividades de teste e as demais fases



Problemas dos modelos sequenciais em geral

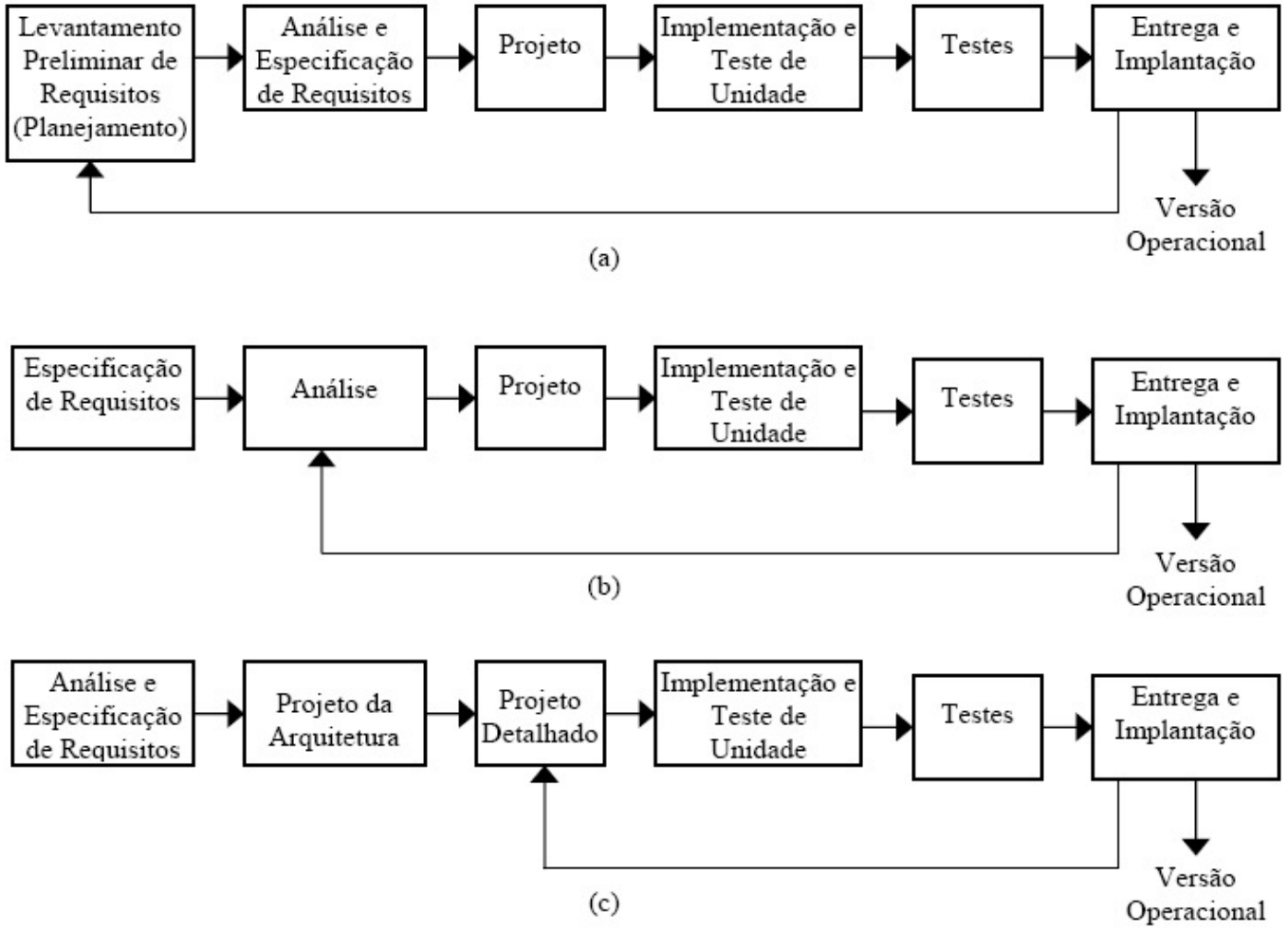
- Projetos reais raramente seguem o fluxo sequencial que o modelo propõe.
- É difícil estabelecer os requisitos de maneira completa, correta e clara no início do projeto.
- Uma versão operacional disponível no final do projeto
- Estados de “bloqueio”no qual alguns membros da equipe precisam esperar que tarefas sejam completas.



Modelo Incremental

- O princípio fundamental é que, a cada ciclo ou iteração, uma versão operacional do sistema será produzida e entregue.
- Requisitos têm de ser minimamente levantados e há de se constatar que o sistema é modular,
- O primeiro incremento tipicamente contém funcionalidades centrais, tratando dos requisitos básicos.
- Comporta diversas variações.

Modelo Incremental



Modelo Incremental Vantagens

- Menor custo e menos tempo são necessários para se entregar a primeira versão;
- Os riscos associados ao desenvolvimento de um incremento são menores, devido ao seu tamanho reduzido;
- O número de mudanças nos requisitos antes da implementação tende a diminuir devido ao curto tempo de desenvolvimento de um incremento.
- É particularmente útil para lidar com riscos técnicos, tal como a adoção de uma nova tecnologia.

Modelo Incremental
Desvantagens

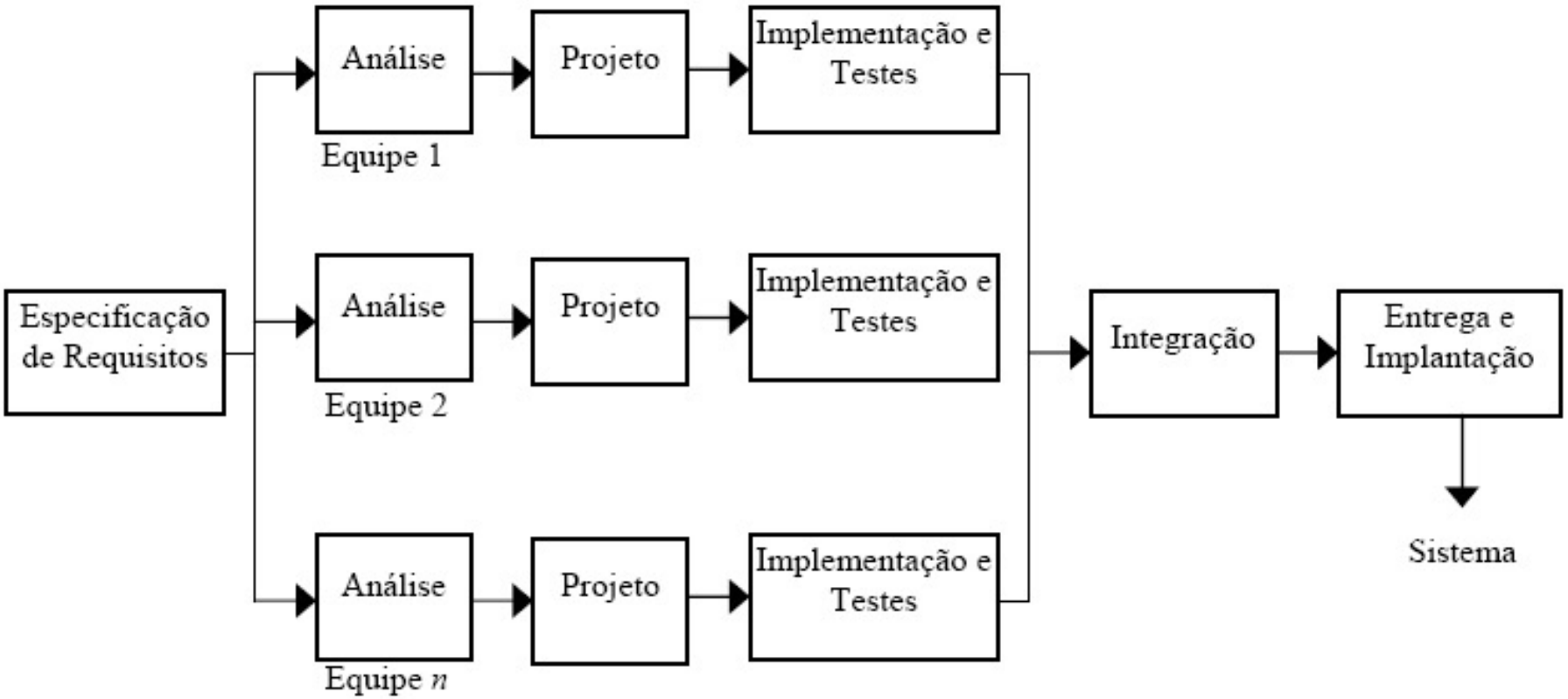
– Se os requisitos não são tão estáveis ou completos quanto se esperava, alguns incrementos podem ter de ser bastante alterados;

– A gerência do projeto é mais complexa, sobretudo quando a divisão em subsistemas inicialmente feita não se mostrar boa.

Modelo RAD (Rapid Application Development)

- Tipo de modelo incremental que prima por um ciclo de desenvolvimento curto (tipicamente de até 90 dias).
- A diferença marcante para o incremental clássico é que os incrementos são desenvolvidos em paralelo por equipes distintas e apenas uma única entrega é feita.

Modelo RAD (Rapid Application Development)



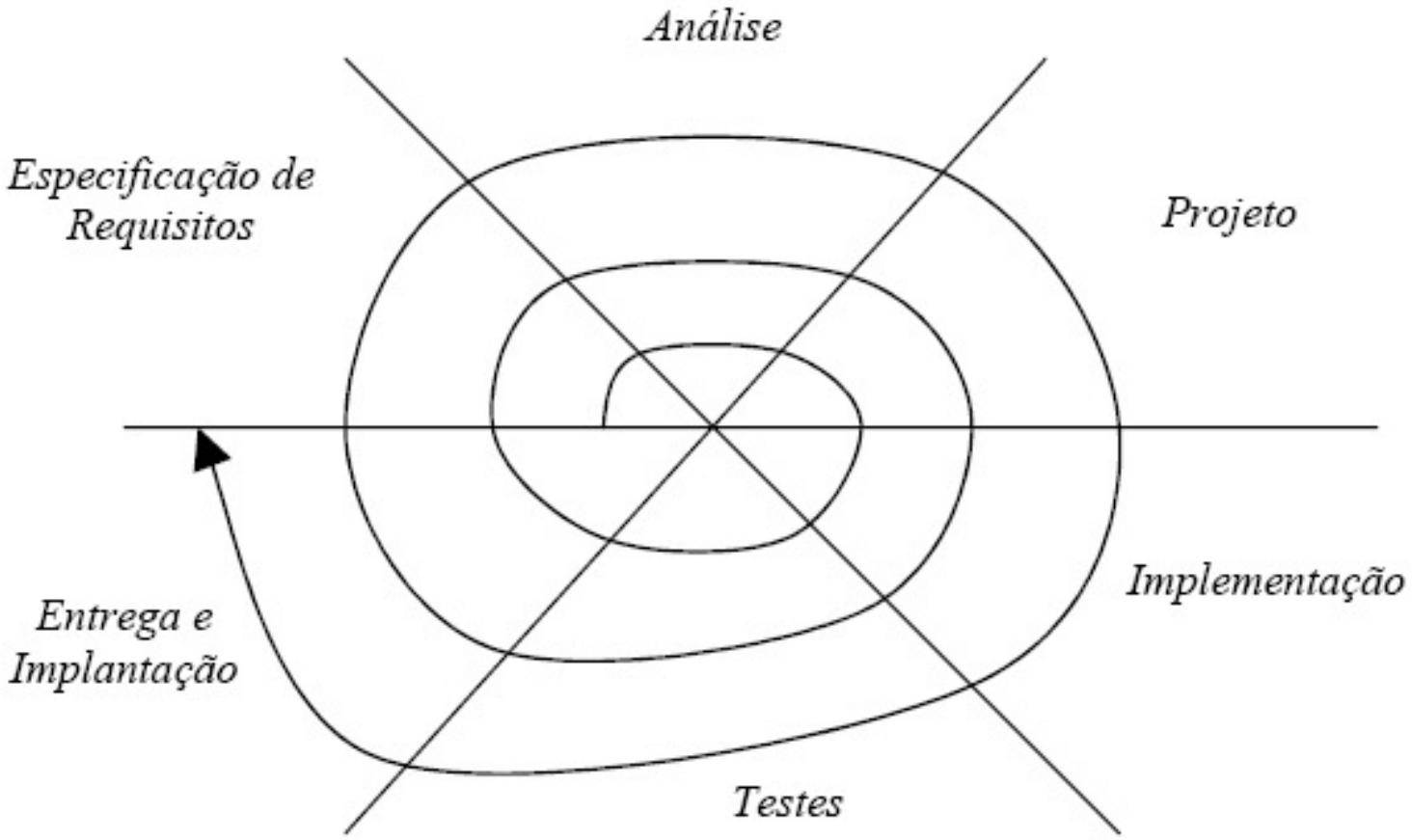
Modelo RAD (Rapid Application Development)

- Os requisitos têm de ser bem-definidos, o escopo do projeto tem de ser restrito e o sistema modular.
- Necessita de recursos humanos para alocar diversas equipes
- Em projetos muito grandes o número de equipes pode crescer demais e a atividade de integração se tornar muito complexa

Modelos evolutivos

- Partem da premissa que o sistema evolui ao longo do tempo.
- É iterativo como os modelos incrementais mas, não há a preocupação de haver entregas de versões operacionais a cada ciclo.
- À medida que o desenvolvimento avança e os requisitos vão ficando mais claros e estáveis, protótipos vão dando lugar a versões operacionais, até que o sistema completo seja construído.

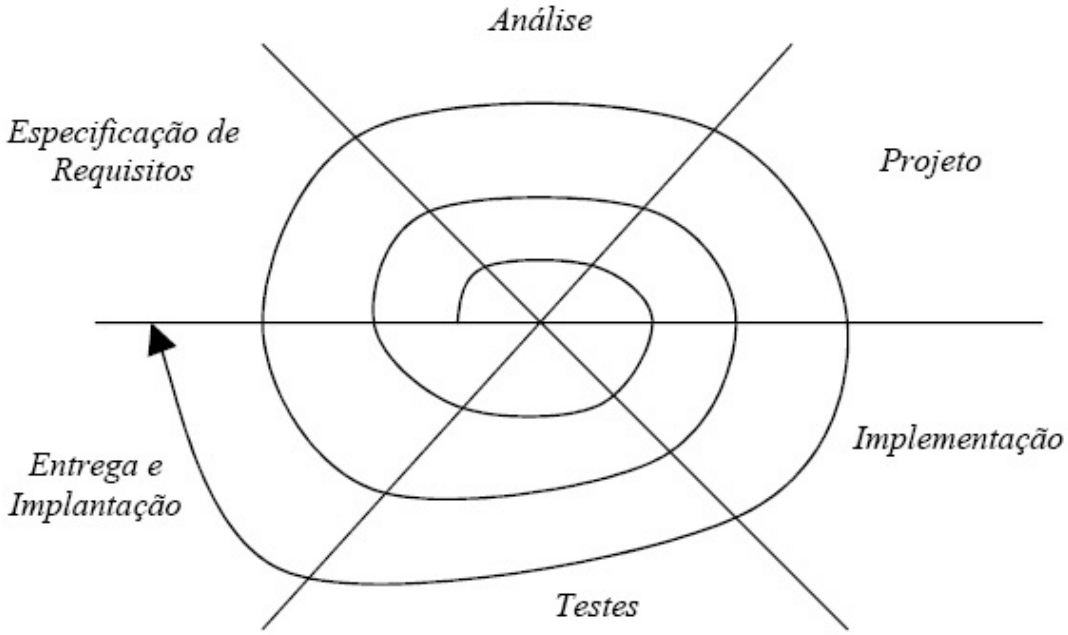
Modelo Evolutivo em Espiral



Modelo em Espiral

- Nos primeiros ciclos nem sempre todas as atividades são realizadas.

- A cada ciclo, o planejamento deve ser revisto com base no feedback do cliente, ajustando, inclusive, o número de iterações planejadas.



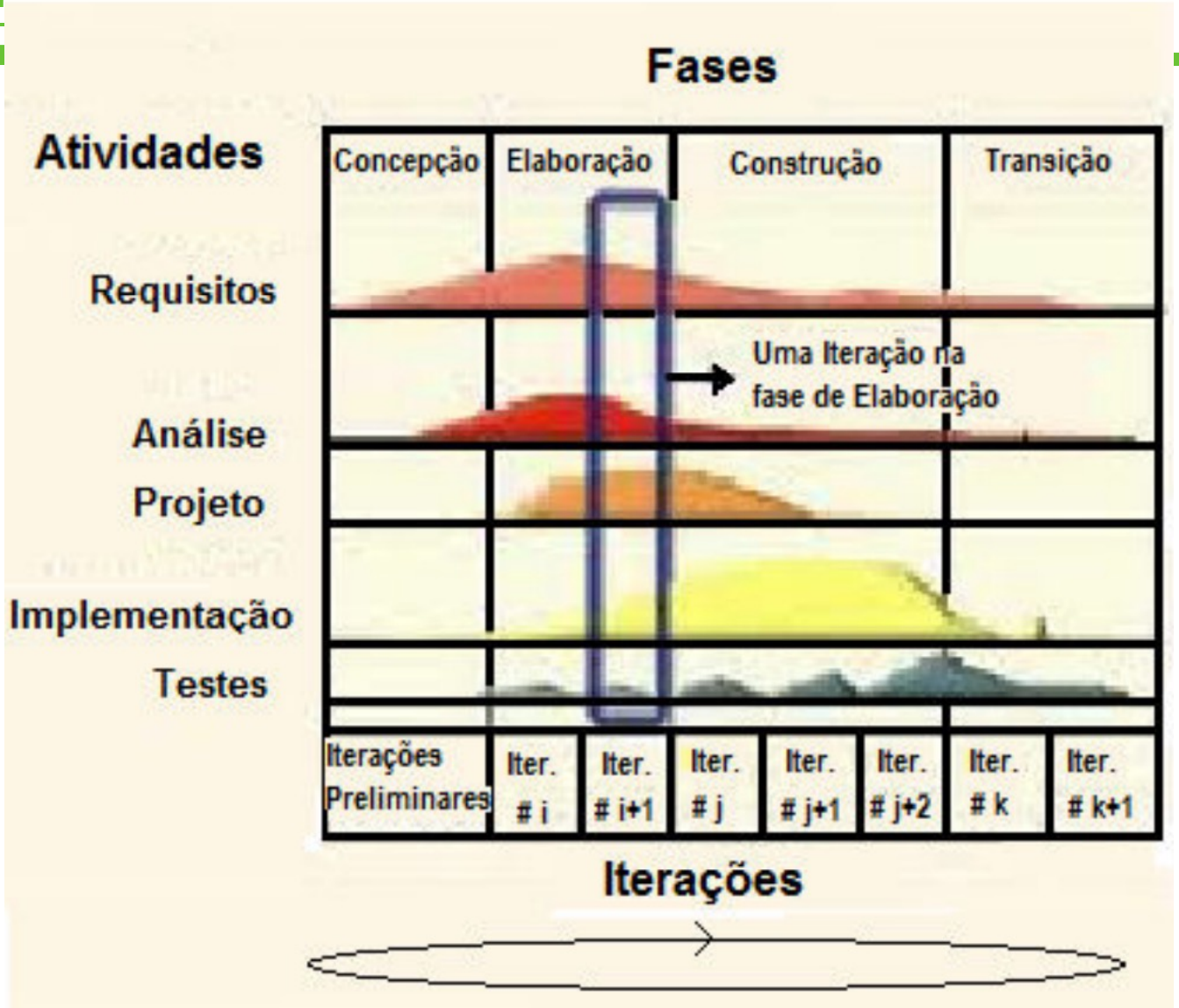
Modelo em Espiral

- Útil quando o problema não é bem definido e ele não pode ser totalmente especificado no início do desenvolvimento;
- Necessária uma forte gerência do projeto e de configuração.
- Pode ser difícil convencer clientes, especialmente em situações envolvendo contrato, que a abordagem evolutiva é gerenciável

Modelo do Processo Unificado (RUP)

- O RUP é mais que um modelo de processo. Inclui também definição detalhada de responsabilidades (papéis), atividades, artefatos e fluxos de trabalho, dentre outros.
- O modelo do RUP é iterativo e evolutivo.
- Organizado em duas dimensões: Fases e atividades.
- As atividades do processo de desenvolvimento são distribuídas ao longo de uma iteração, em função do foco da fase correspondente.

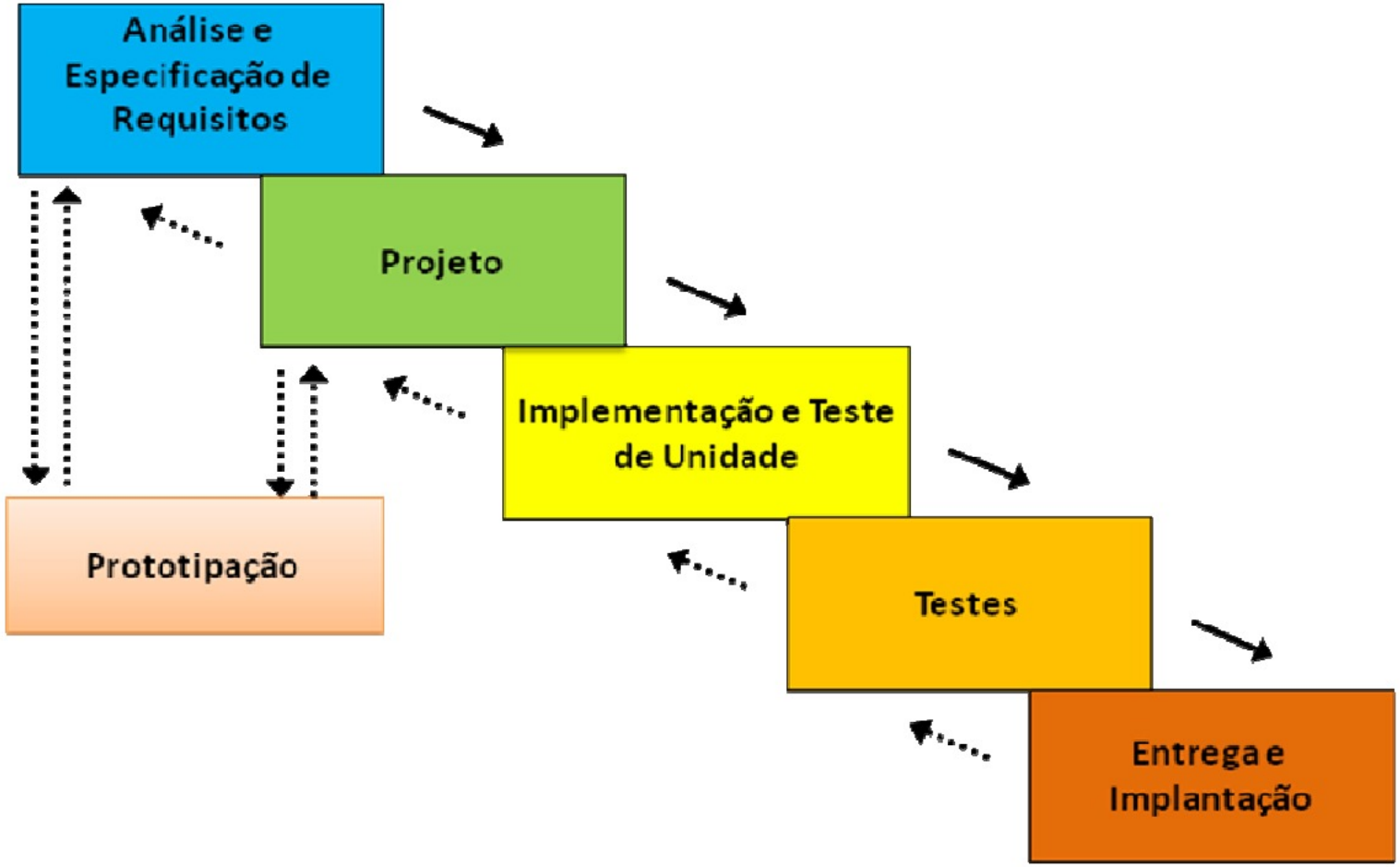
Modelo do Processo Unificado (RUP)



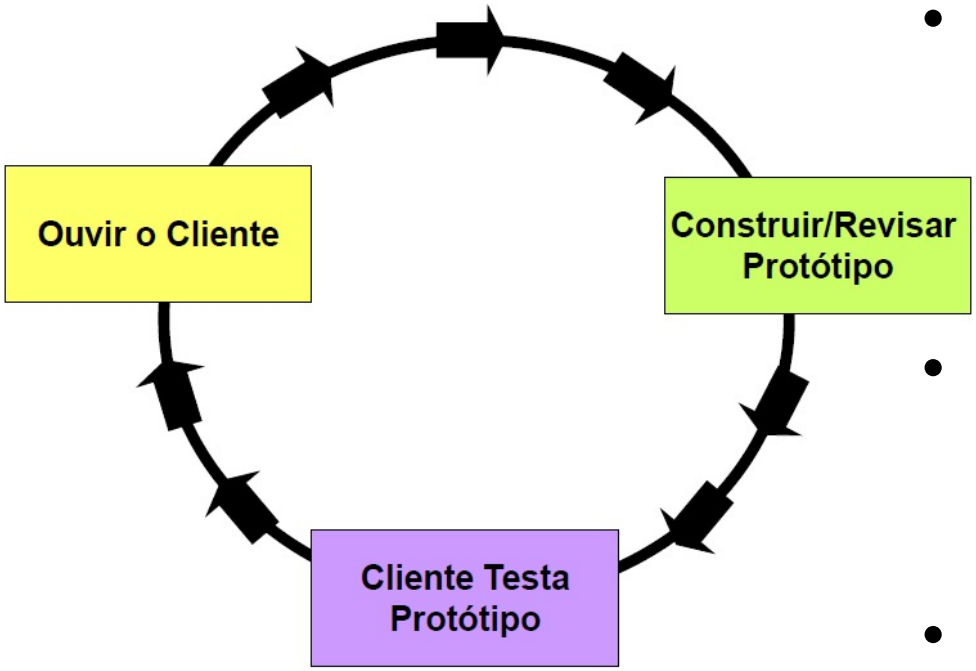
Prototipação

- Técnica para ajudar engenheiros de software e clientes a entender o que está sendo construído quando os requisitos não estão claros.
- Apesar de se encaixarem perfeitamente em modelos evolutivos, podem ser aplicados no contexto de qualquer modelo de processo

Prototipação



Modelo Prototipação



- Muito arriscado. Só aplicável a sistemas simples.

- Modelo iterativo em que projetos descartáveis e que evoluem até se tornar o sistema real.
- Interessados podem achar que o protótipo é a versão final do software.
- O engenheiro assume o compromisso de o protótipo entrar em funcionamento rapidamente

Resumo

- Nesta aula vimos:
  - A necessidade da engenharia de software
  - As atividades típicas do processo de software
  - Os modelos de ciclo de vida de software