



**INSTITUTO
FEDERAL**

Espírito Santo

Campus
Colatina

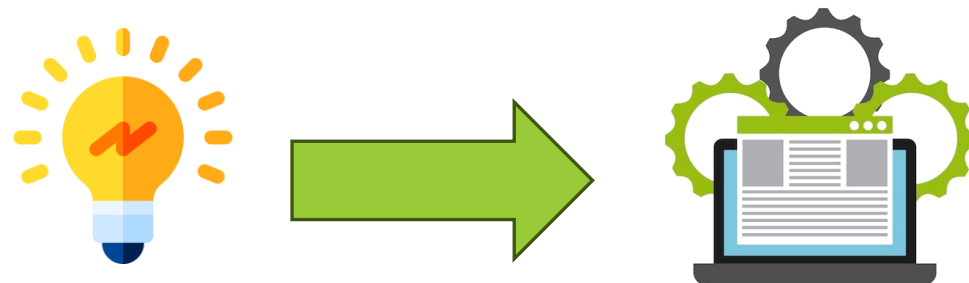
Análise de Sistemas

TEMA: MODELOS DE CICLO DE VIDA DE SOFTWARE

PROFESSOR: ALLAN FERNANDES BALARDINO

Ciclo de Vida do Software

- O início de todo projeto de software acontece com uma ideia na mente de alguém;
- Para construir a realização desta ideia, a equipe de desenvolvimento cria uma série de modelos conceituais distribuídos entre atividades que transforma a ideia em software;



Modelos de Ciclo de Vida

- Mas como as atividades são encadeadas dentro do processo?
- O modelo de ciclo de vida de software:
 - É um “roteiro” que ajuda a percorrer uma série de passos previsíveis, mostrando quando devem ser executados e suas dependências;



Objetivo do Ciclo de Vida

- Fornecer uma estrutura bem definida para prover estabilidade, controle e organização no que diz respeito ao desenvolvimento de software, proporcionando melhorias em:
 - Organização e planejamento;
 - Gerenciamento de riscos;
 - Qualidade e consistência;
 - Previsibilidade e Controle de Custos;
 - Melhoria contínua e documentação;

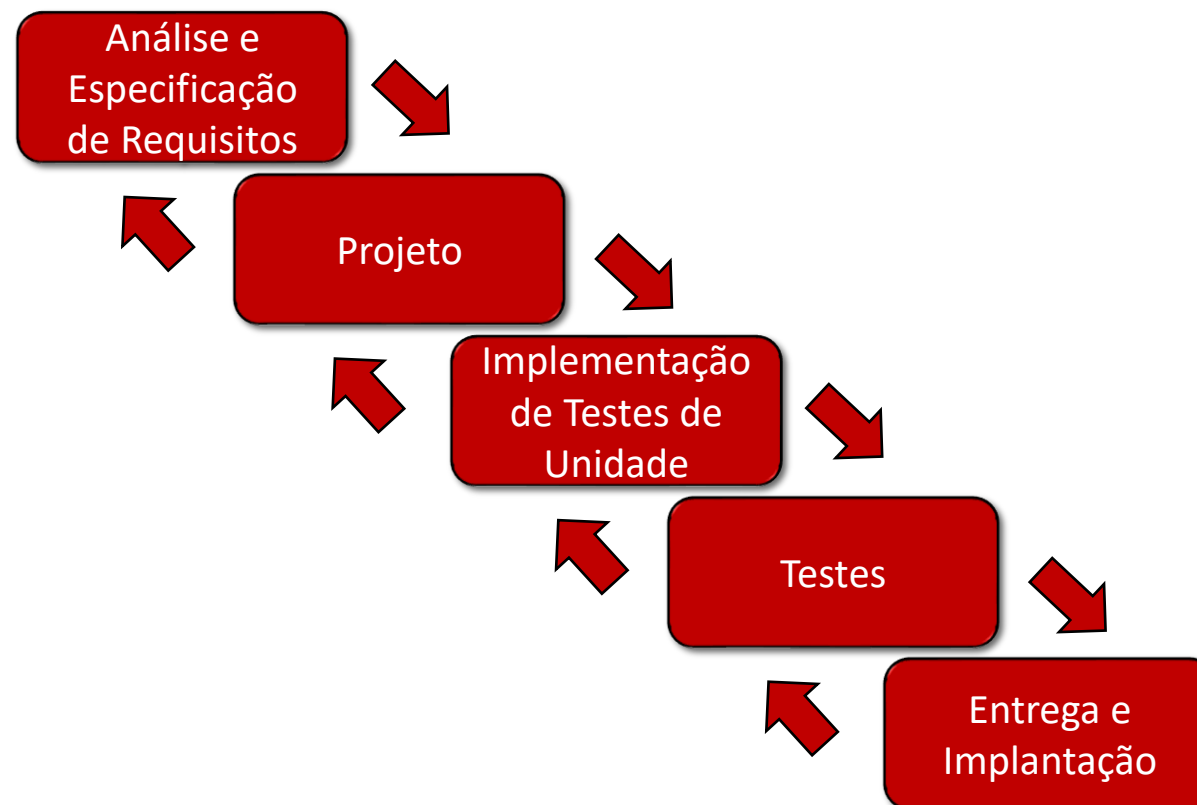
Modelos do Ciclo de Vida

- Sequenciais:
 - Modelo em cascata;
 - Modelo em V;
- Incrementais:
 - Modelo incremental;
 - Modelo RAD (*Rapid Application Development*);
- Evolutivos:
 - Modelo em Espiral;
 - Modelo do Processo Unificado (RUP);

Modelos Sequenciais (Cascata)

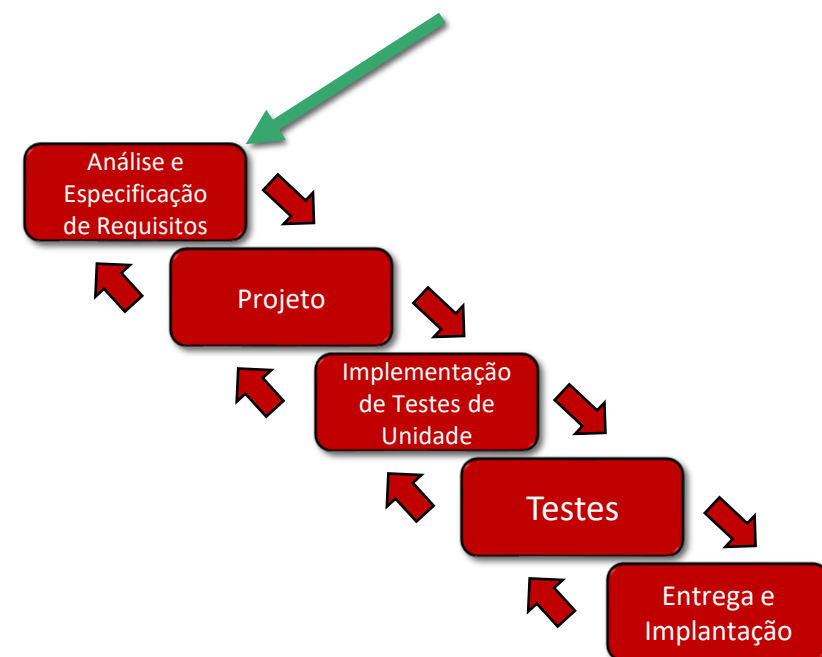
- Primeiro modelo de ciclo de vida criado:
 - Criado em 1970 por Winston W. Royce;
 - Abordar a complexidade do software de maneira sequencial, dividindo o processo em fases claras e estruturadas.
 - Uma fase só inicia com a conclusão da etapa anterior;
 - É permitido retorno à fase anterior para correção;
 - Cada fase serve como base aprovada e documentada para o passo seguinte, facilitando a gerência de configuração.

Modelos Sequenciais (Cascata - Fases)



Modelos Sequenciais (Cascata - Fases)

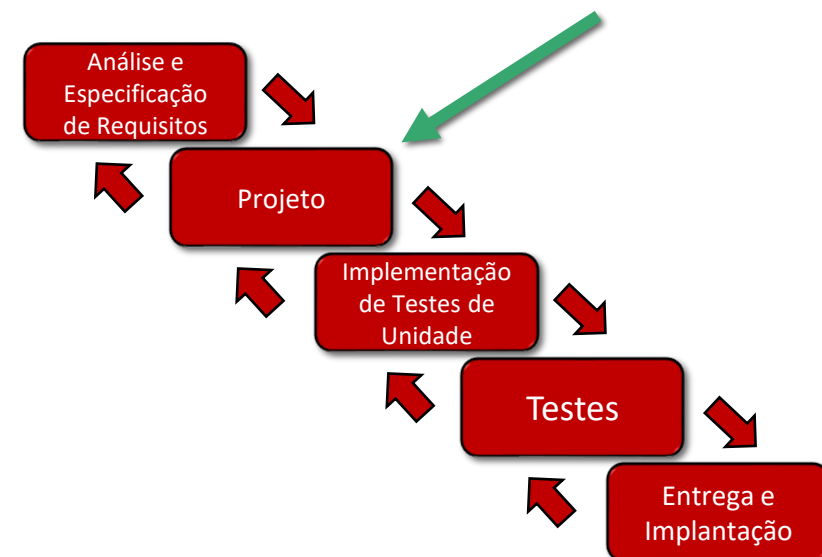
- **Fase: Análise e Especificações de Requisitos**
 - Nesta fase são estabelecidas as funções, restrições e objetivos do sistema;
 - Fase em que o analista compreende o domínio do problema proposto (regras de negócio);
 - Os requisitos devem ser documentados e revistos com o cliente;
 - Levantados requisitos mais amplos, como: hardware, pessoas e softwares necessários;



Modelos Sequenciais (Cascata - Fases)

- **Fase: Projeto**

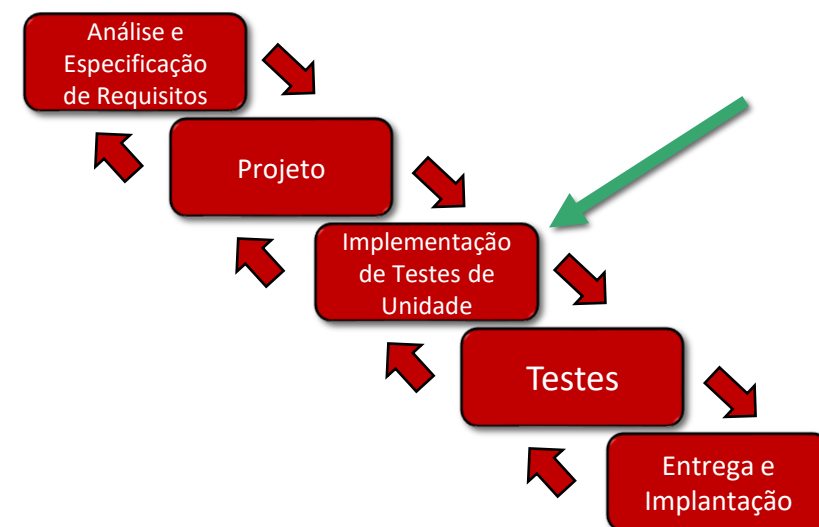
- Identificação e descrição das abstrações fundamentais, como:
 - Estrutura de dados;
 - Planejamento de base de dados;
 - Diagramas UML;
 - Documentação de especificações técnicas;
 - Arquitetura de software;
 - Caracterização de interface;



Modelos Sequenciais (Cascata - Fases)

○ Fase: Implementação

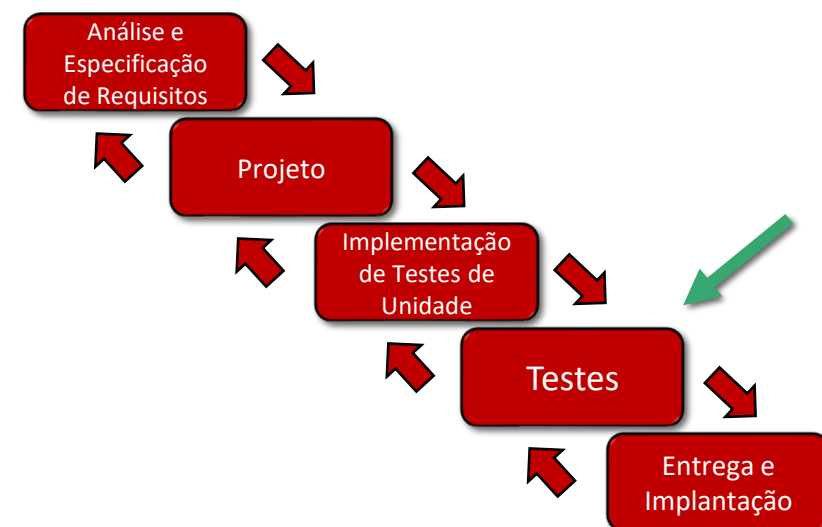
- Tradução das regras de negócio em notação “entendível” por máquinas;
- Se o projeto foi executado detalhadamente, onde todos os procedimentos e regras foram bem definidos e explicados, a codificação pode ser executada muitas vezes até mecanicamente;
- Comum a contratação de bons profissionais para análise e projeto (alto detalhamento) e terceirização da implementação;



Modelos Sequenciais (Cascata - Fases)

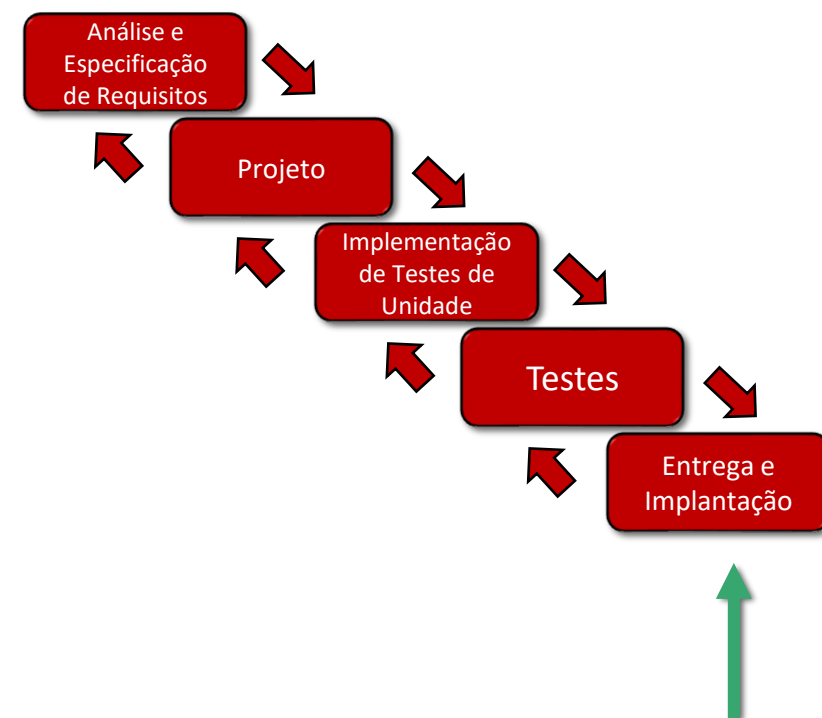
- **Fase: Testes**

- Testes sobre aspectos internos:
 - Garante que todas as instruções lógicas do sistema tenham sido executadas, ou seja, não há erro de código;
- Testes sobre aspectos externos:
 - Confirmar se os dados inseridos geram os dados esperados, conforme requisitado pelo cliente;



Modelos Sequenciais (Cascata - Fases)

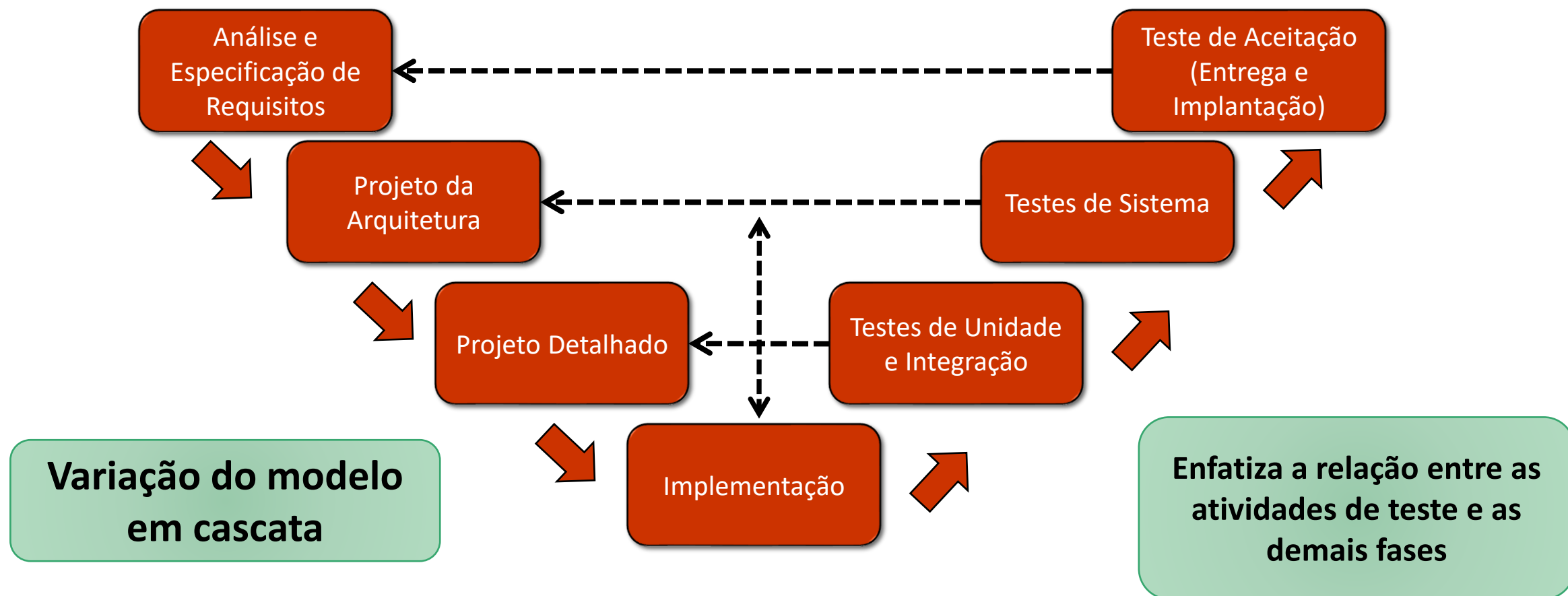
- **Fase: Entrega e Implementação**
 - Mesmo depois do software entregue ao cliente, haverá mudanças;
- Possíveis motivos:
 - Erros encontrados;
 - Adaptações para atender ao ambiente do cliente;
 - Exigência de acréscimos funcionais;
 - Desempenho insatisfatório
- **Para cada manutenção, todo o ciclo de vida é reexecutado!**



Modelos Sequenciais (Modelo em V)

- Variação do modelo cascata;
- Semelhanças:
 - Sequenciais, com fases claras e estruturadas, com cada uma tendo que ser concluída antes de iniciar a próxima;
- Diferenças:
 - Etapas de teste correspondente a cada etapa do ciclo;
 - Busca da garantia da qualidade ao fim de cada fase;

Modelos Sequenciais (Modelo em V)



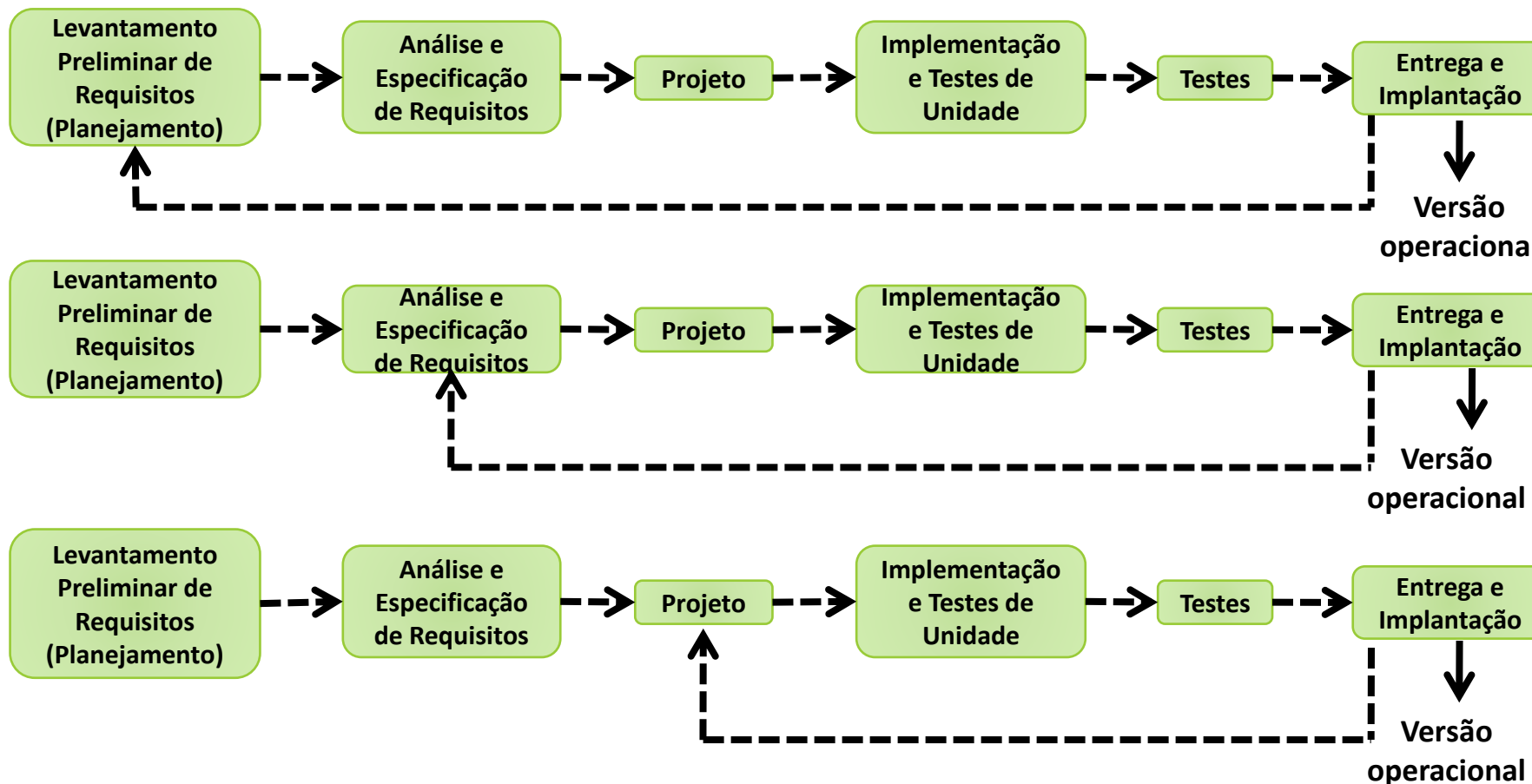
Problemas dos Modelos Sequenciais

- Projetos reais raramente seguem o fluxo sequencial que o modelo propõe;
- Difícil para o cliente declarar todas as exigências (requisitos) explicitamente em uma única fase;
- Uma versão do sistema só estará disponível em um momento tardio do cronograma, contrariando expectativa do cliente;
- Pode ocorrer estados de bloqueio no qual alguns membros da equipe precisam esperar que tarefas estejam completas:
 - Desenvolvedores aguardando time de design;
 - Testadores aguardando liberação por parte dos desenvolvedores;
- **Exceção:** projetos pequenos e de requisitos bem definidos;

Modelos Incrementais (Incremental Clássico)

- O princípio fundamental é que, a cada ciclo ou iteração, uma versão operacional do sistema será produzida e entregue;
- Os requisitos são minimamente levantados e agrupados para se tornar um sistema modular:
 - Os detalhamentos dos requisitos são realizados nos incrementos;
- O primeiro incremento tipicamente contém funcionalidades centrais, tratando dos requisitos básicos;

Modelos Incrementais (Incremental Clássico)



Modelo incremental clássico: **Vantagens**

- Menor custo e menos tempo serão necessários para se entregar uma primeira versão;
- Os riscos associados ao desenvolvimento de um incremento são menores devido ao tamanho reduzido da funcionalidade;
- O número de mudanças nos requisitos antes da implementação tende a diminuir devido ao curto tempo de desenvolvimento de um incremento;

Modelo incremental clássico: **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 não seja feita de maneira eficiente:
 - Pode ser liberada uma funcionalidade que depende de outra que ainda não foi disponibilizada, gerando atraso;
 - Exige planejamento contínuo, integração frequente e mudanças mais constantes;

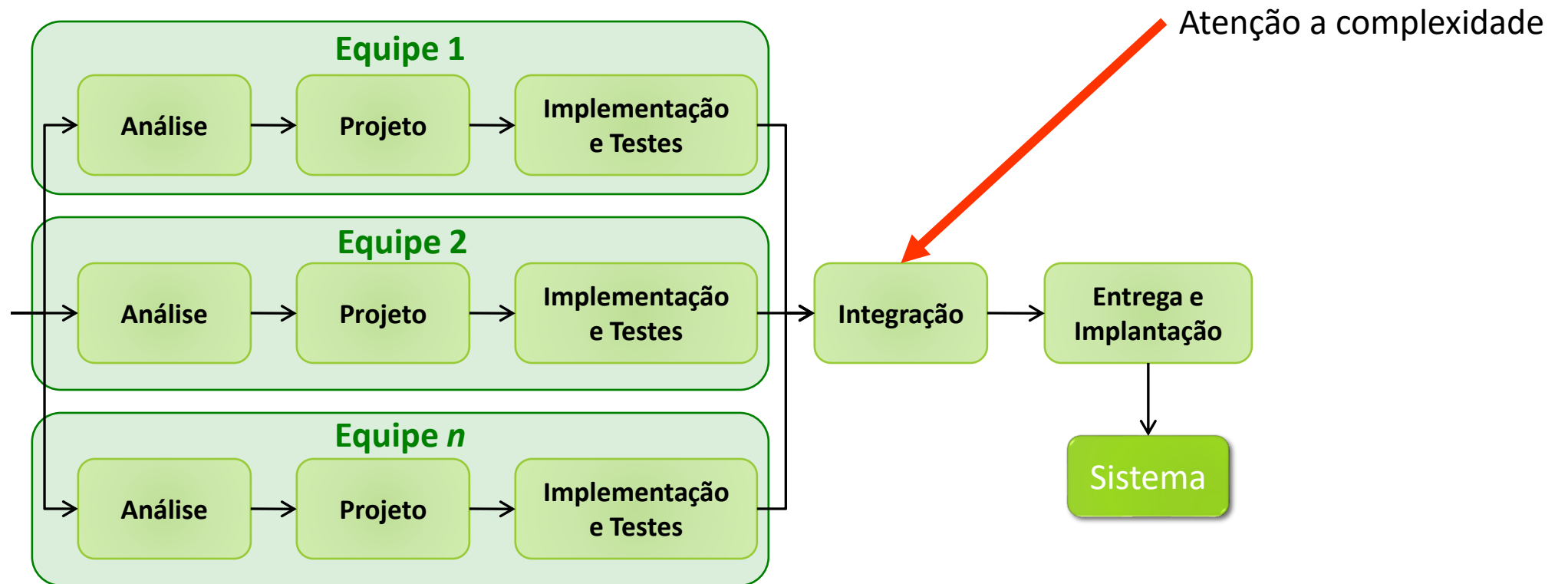
Modelos Incrementais

(RAD – Rapid Application Development)

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

Modelos Incrementais

(RAD – Rapid Application Development)

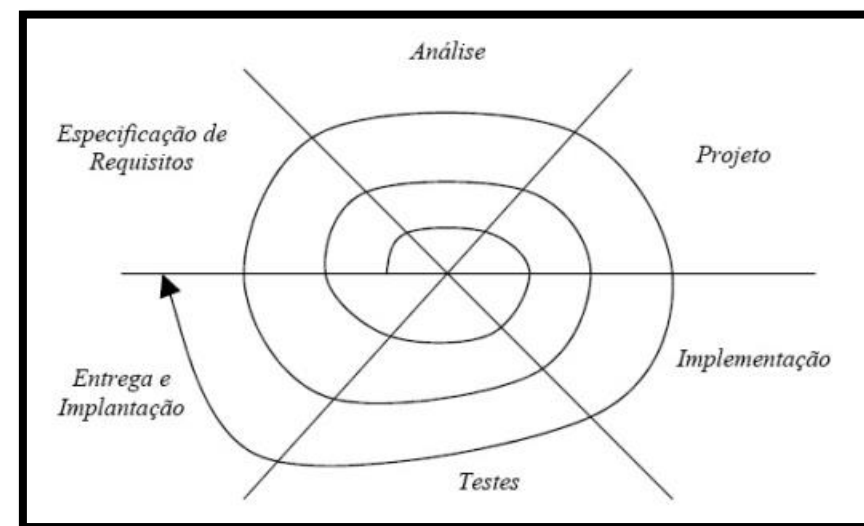


Modelos Evolutivos

- Modelos incrementais pressupõem requisitos bem definidos, o que muitas vezes não acontece;
- **Modelos evolutivos:**
 - Partem da premissa que o sistema evolui ao longo do tempo;
 - É iterativo como os incrementais mas não há preocupação de haver entregas de versões a cada ciclo;
 - A medida que o desenvolvimento avança e os requisitos vão ficando mais claros e estáveis, **versões intermediárias** (não operacionais) vão dando lugar a versões operacionais, até que o sistema seja de fato, finalizado;

Modelos Evolutivos (Espiral)

- Nos primeiros ciclos nem sempre todas as atividades são realizadas:
 - Por exemplo, o produto resultante do primeiro ciclo pode ser uma especificação do produto ou um estudo de viabilidade;
- A cada ciclo, o planejamento deve ser revisto com base no feedback do cliente, ajustando, inclusive o número de iterações inicialmente planejadas;



Modelos Evolutivos (Espiral)

- Útil quando o problema não é bem definido e não pode ser totalmente especificado no início do desenvolvimento;
- Necessita uma forte gerência do projeto e de configuração;
- Dificuldade de convencer clientes a iterações sem versão operacional (sobretudo em situações envolvendo contratos de software);

Incremental x Evolutivo (exemplo sistema acadêmico básico)

Desenvolvimento Incremental

Incremento 1: Cadastro de Estudantes e Professores

Funcionalidade: Permitir o cadastro de estudantes e professores no sistema.

Entrega: Uma versão funcional onde é possível adicionar, editar e visualizar estudantes e professores.

Incremento 2: Gestão de Disciplinas

Funcionalidade: Adicionar a funcionalidade de cadastro e gerenciamento de disciplinas.

Entrega: Sistema atualizado que agora inclui o gerenciamento de disciplinas, além das funcionalidades do Incremento 1.

Incremento 3: Matrículas

Funcionalidade: Implementar a funcionalidade de matrícula de estudantes nas disciplinas.

Entrega: Sistema que inclui cadastro de estudantes e professores, gerenciamento de disciplinas e funcionalidade de matrículas.

Incremento 4: Lançamento de Notas

Funcionalidade: Adicionar a funcionalidade de lançamento e gerenciamento de notas dos estudantes.

Entrega: Sistema que inclui todas as funcionalidades anteriores mais o lançamento de notas.

Incremento 5: Relatórios e Estatísticas

Funcionalidade: Adicionar geração de relatórios e estatísticas acadêmicas.

Entrega: Sistema completo com todas as funcionalidades anteriores e a geração de relatórios.

Desenvolvimento Evolutivo

Iteração 1: Protótipo Inicial

Funcionalidade: Desenvolver um protótipo básico com o cadastro de estudantes e uma interface simples para feedback inicial.

Entrega: Protótipo funcional que permite adicionar e visualizar estudantes.

Iteração 2: Expansão do Protótipo

Funcionalidade: Melhorar o cadastro de estudantes e adicionar o cadastro de professores com base no feedback anterior.

Entrega: Protótipo melhorado que agora inclui estudantes e professores.

Iteração 3: Gestão de Disciplinas

Funcionalidade: Adicionar uma interface básica para cadastro e gerenciamento de disciplinas com melhorias incrementais.

Entrega: Protótipo atualizado com funcionalidades de gerenciamento de disciplinas.

Iteração 4: Matrículas

Funcionalidade: Implementar funcionalidades básicas de matrícula de estudantes nas disciplinas, incorporando melhorias contínuas.

Entrega: Protótipo que inclui cadastro de estudantes e professores, gerenciamento de disciplinas e matrículas.

Iteração 5: Lançamento de Notas

Funcionalidade: Adicionar lançamento de notas, fazendo ajustes finos em todas as funcionalidades.

Entrega: Protótipo evoluído com todas as funcionalidades básicas e melhorias contínuas.

Iteração 6: Relatórios e Estatísticas

Funcionalidade: Adicionar geração de relatórios e estatísticas acadêmicas.

Entrega: Sistema que evoluiu ao longo de várias iterações e agora inclui todas as funcionalidades necessárias.

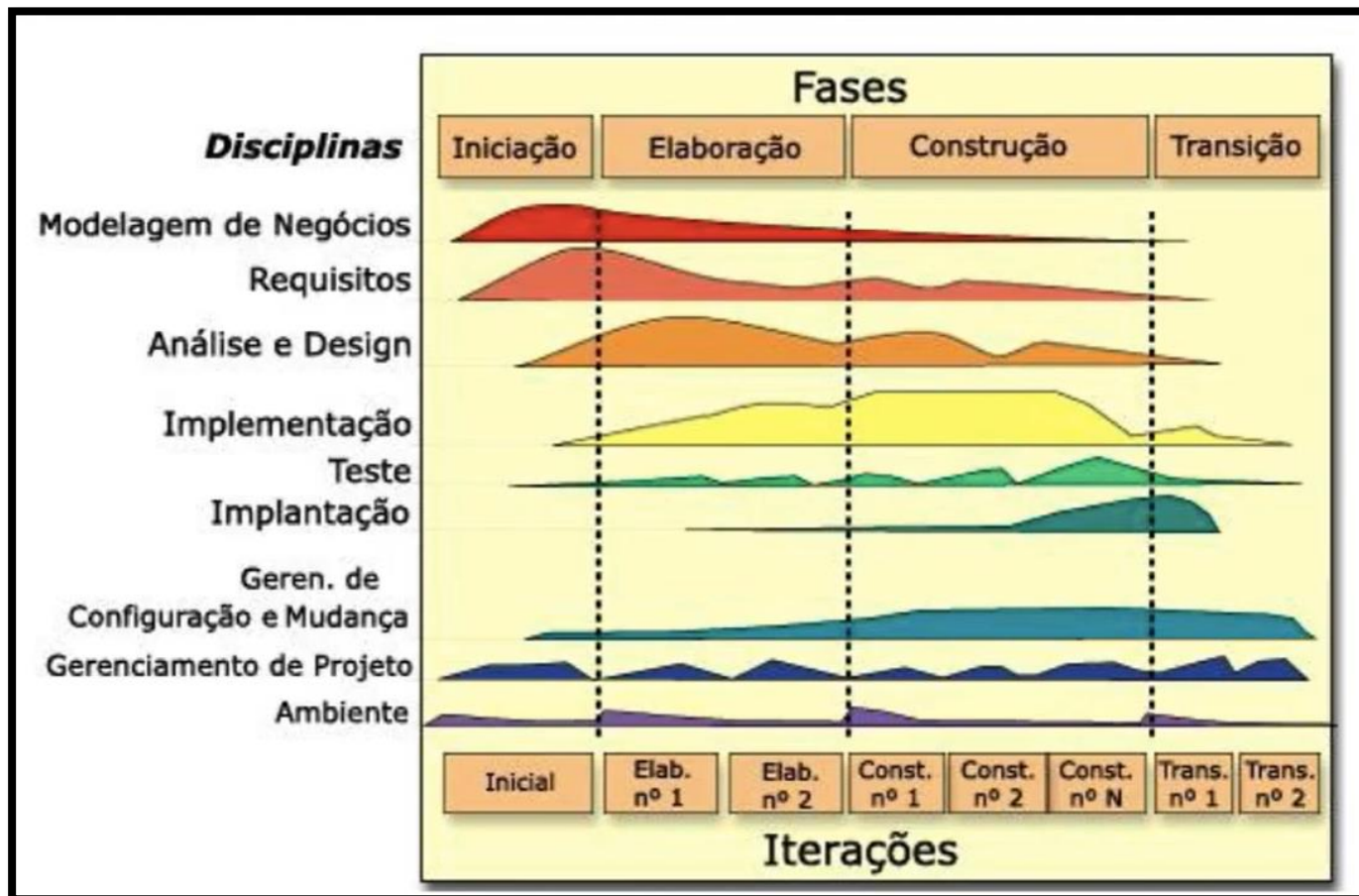
Modelos Evolutivos

(*RUP – Rational Unified Process*)

- Modelo do processo unificado da *Rational* (empresa hoje pertencente a IBM):
 - Além de definir o processo em si, o faz para definição de responsabilidades, atividades, artefatos, fluxos de trabalho, entre outros);
- Organizado em 2 dimensões (4 fases e 9 disciplinas, sendo essas as áreas dentro do processo);
- Iterações ocorre ao longos das fases e também nas disciplinas;

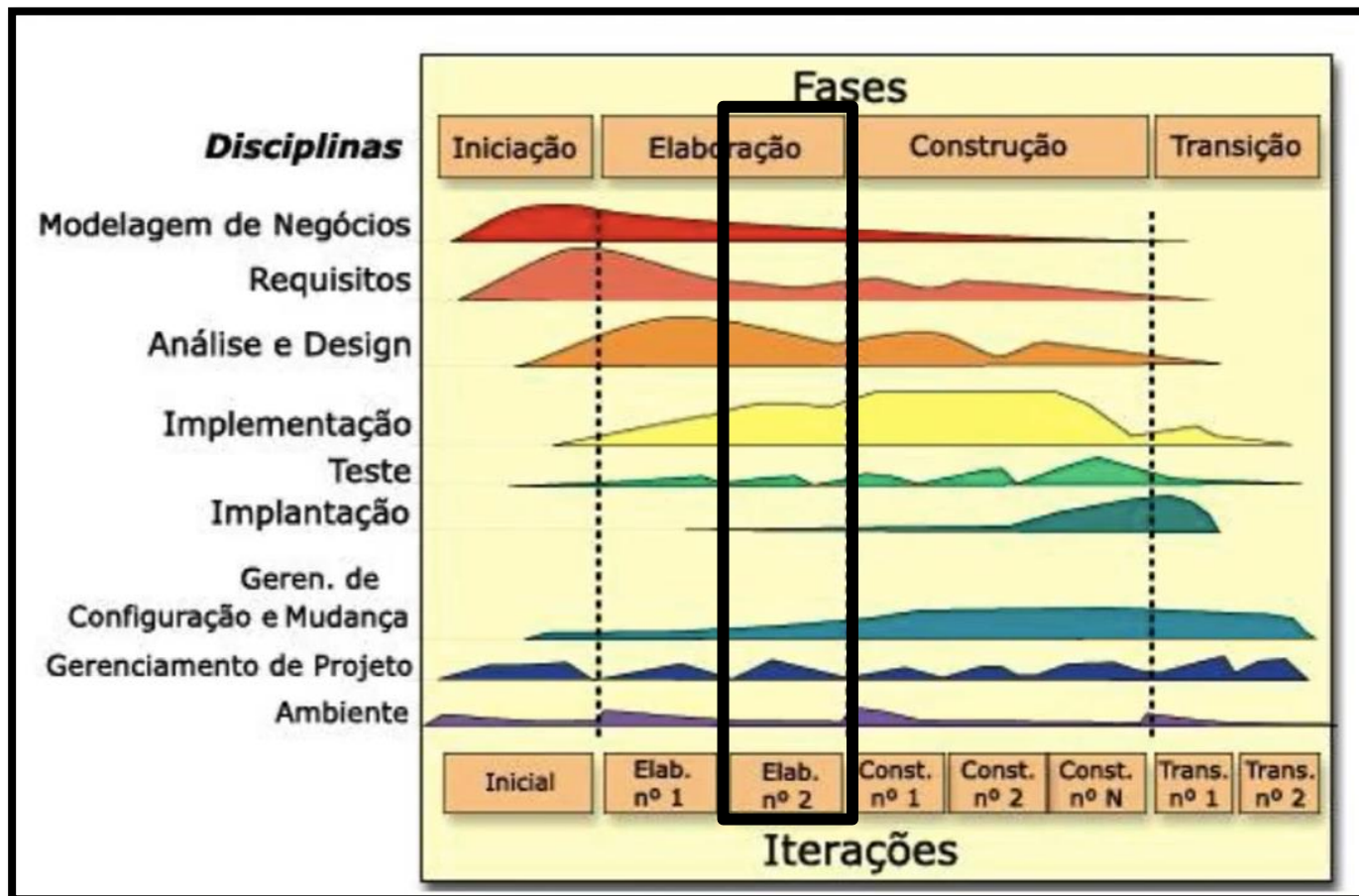
Modelos Evolutivos

(RUP – Rational Unified Process)



Modelos Evolutivos

(RUP – Rational Unified Process)



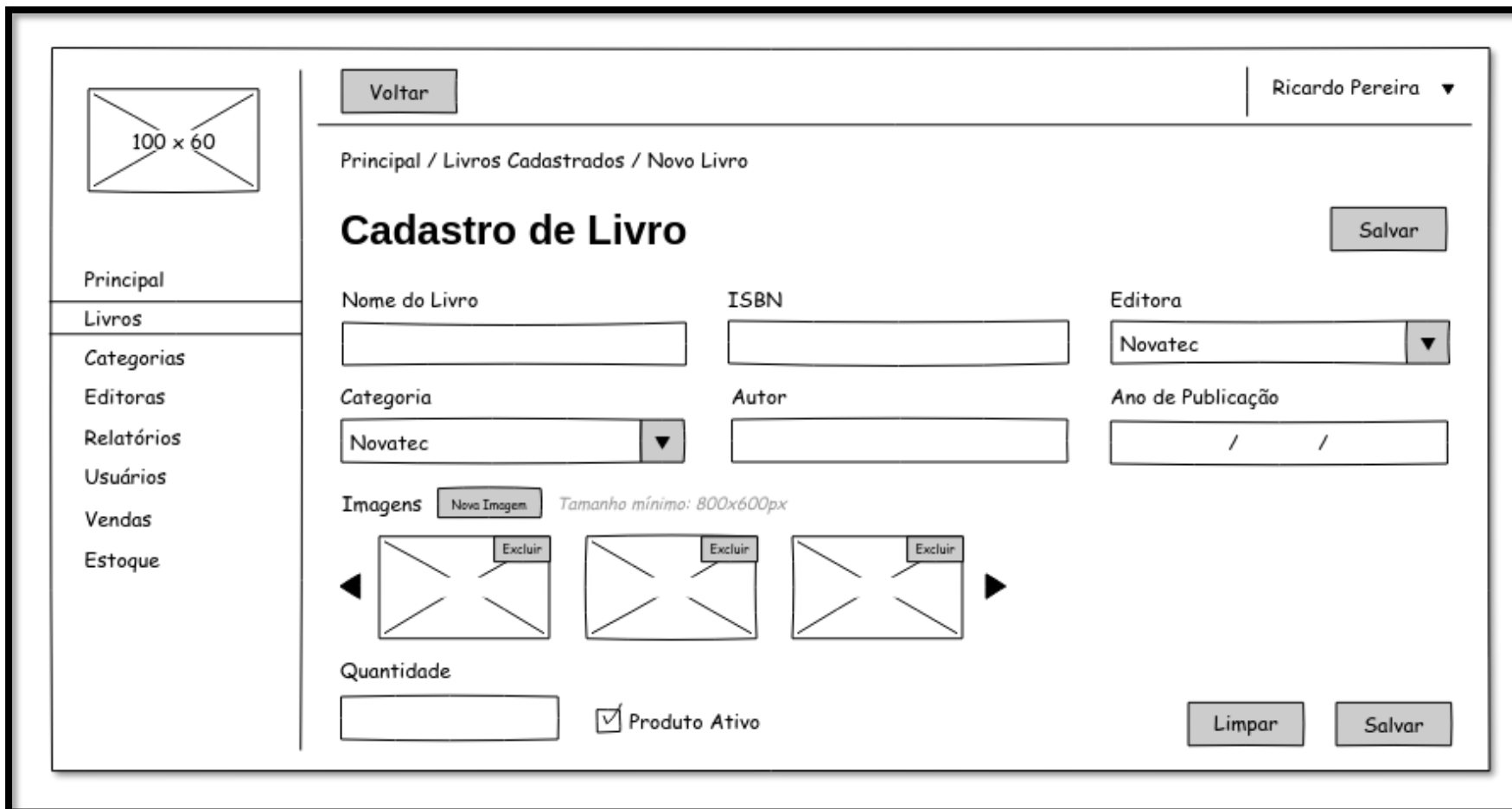
Modelos Evolutivos (Fases do RUP)

- **Iniciação:** foco em um bom entendimento do problema e levantamento de requisitos;
- **Elaboração:** os objetivos desta fase são analisar o domínio do problema, estabelecer a arquitetura do sistema, refinar o plano do projeto e identificar seus maiores riscos. Assim, em termos do processo de desenvolvimento, o foco são as atividades de análise e projeto;
- **Construção:** envolve o projeto detalhado de componentes, sua implementação e testes. Nesta fase, os componentes do sistema são desenvolvidos, integrados e testados;
- **Transição:** como o próprio nome indica, o propósito desta fase é fazer a transição do sistema do ambiente de desenvolvimento para o ambiente de produção. São feitos testes de sistema e de aceitação e a entrega do sistema aos seus usuários;

Prototipação

- Muitas vezes cliente sabe apenas um conjunto geral de objetivos que o sistema precisa atender, não sendo capaz de identificar claramente funcionalidades e requisitos;
- Etapas de requisitos do modelo podem ajudar, mas pode ocorrer de cliente só ter a dimensão do que está sendo construído quando colocado diante do sistema;
- Desenhos ou “Imitações” do sistema que será construído;
- Pode ser aplicado em etapas dos modelos apresentados;
- **Necessário alinhamento de expectativas com cliente!**

Prototipação



The wireframe shows a web interface for book registration. On the left is a sidebar with a menu: Principal, Livros (highlighted), Categorias, Editoras, Relatórios, Usuários, Vendas, and Estoque. Above the menu is a placeholder for a 100 x 60 image. The main content area has a top bar with a 'Voltar' button and a user profile 'Ricardo Pereira'. Below this is a breadcrumb 'Principal / Livros Cadastrados / Novo Livro' and a title 'Cadastro de Livro' with a 'Salvar' button. The form contains several fields: 'Nome do Livro', 'ISBN', 'Editora' (a dropdown menu with 'Novatec' selected), 'Categoria' (a dropdown menu with 'Novatec' selected), 'Autor', and 'Ano de Publicação' (a date field with slashes). There is an 'Imagens' section with a 'Nova Imagem' button, a note 'Tamanho mínimo: 800x600px', and three image placeholders, each with an 'Excluir' button. At the bottom, there is a 'Quantidade' field, a checked checkbox for 'Produto Ativo', and 'Limpar' and 'Salvar' buttons.

Principal / Livros Cadastrados / Novo Livro

Cadastro de Livro

Nome do Livro **ISBN** **Editora** **Salvar**

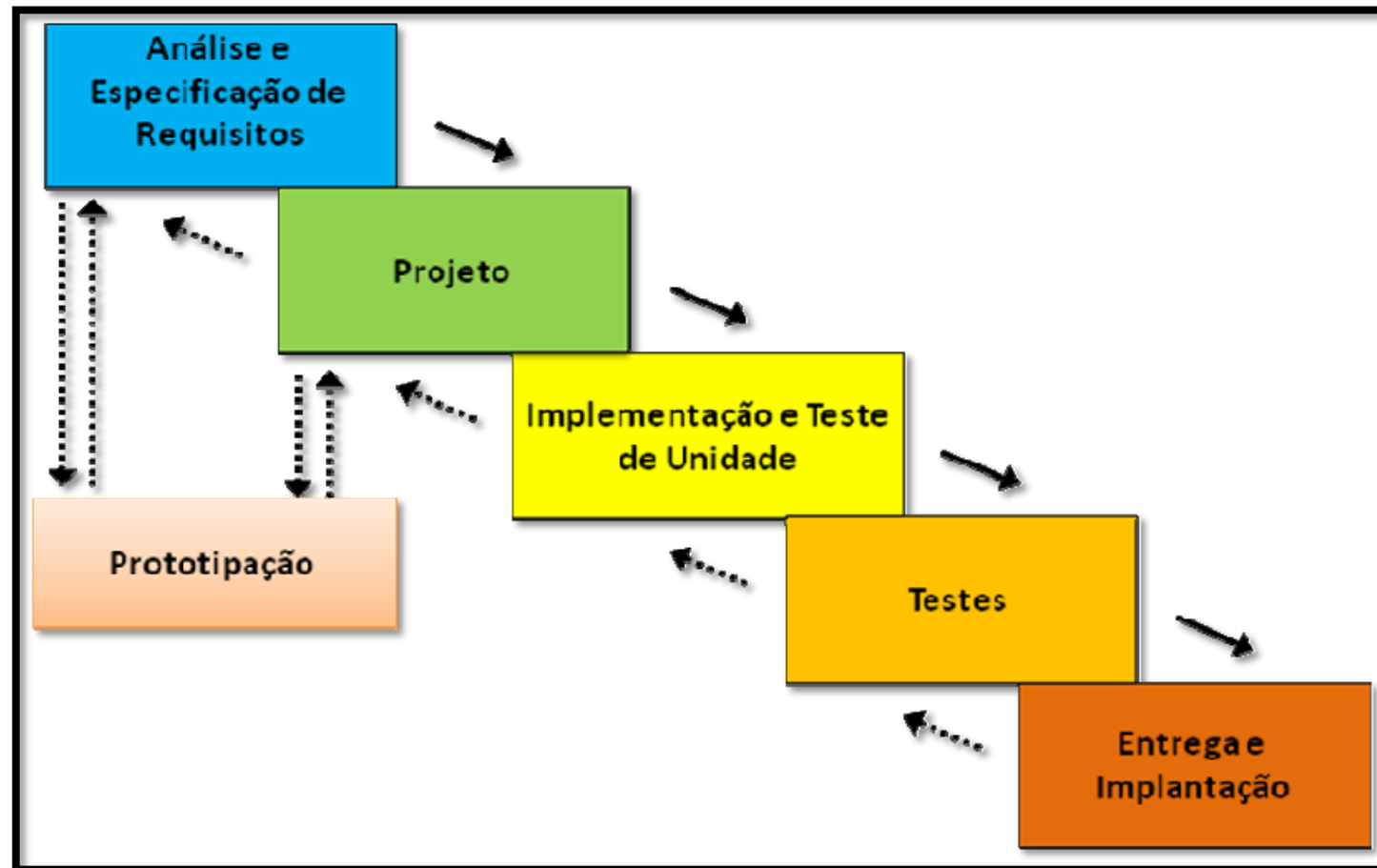
Categoria **Autor** **Ano de Publicação**

Imagens **Nova Imagem** *Tamanho mínimo: 800x600px*

Excluir **Excluir** **Excluir**

Quantidade ☒ **Produto Ativo** **Limpar** **Salvar**

Prototipação (aplicada ao modelo cascata)



Prototipação (**Vantagens**)

- Validação mais eficaz dos requisitos, antecipando expectativas, feedback, economizando tempo e custo;
- Redução de riscos;
- Comunicação clara entre pessoas, buscando entendimento comum;
- Pode servir para surgimento de novas ideias;

Prototipação (**Desvantagens**)

- Usuários podem interpretar como o produto final:
 - Pressão por entrega do produto;
 - Pode ocorrer quebra de expectativa na entrega do software;
- Difícil medir questões como performance, segurança, situações que envolvem integração com outros sistemas...;

Referências

Este material foi baseado no
produzido pelo professor
Victorio Albani Carvalho