Engenharia de Software I

Prof. Agnaldo Cieslak, MsC

RUP

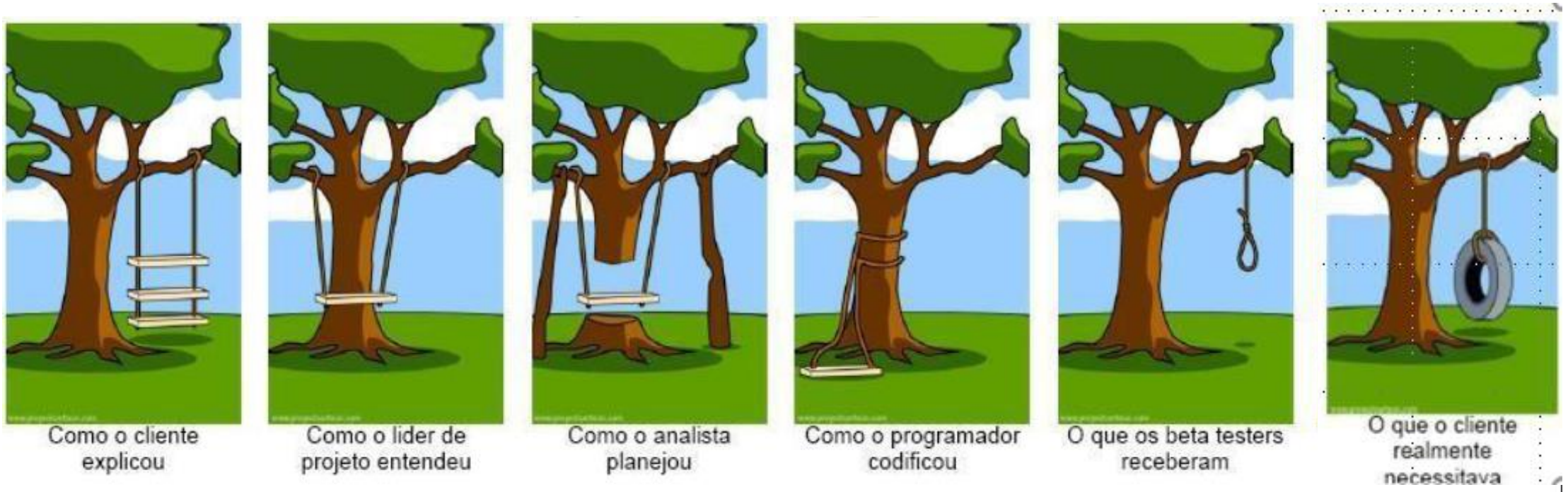
Rational Unified Process (RUP)

Desenvolvimento Tradicional

Tradicionalmente, os projetos são organizados para percorrer cada disciplina em sequência, uma e apenas uma vez. Isso leva ao ciclo de vida em cascata

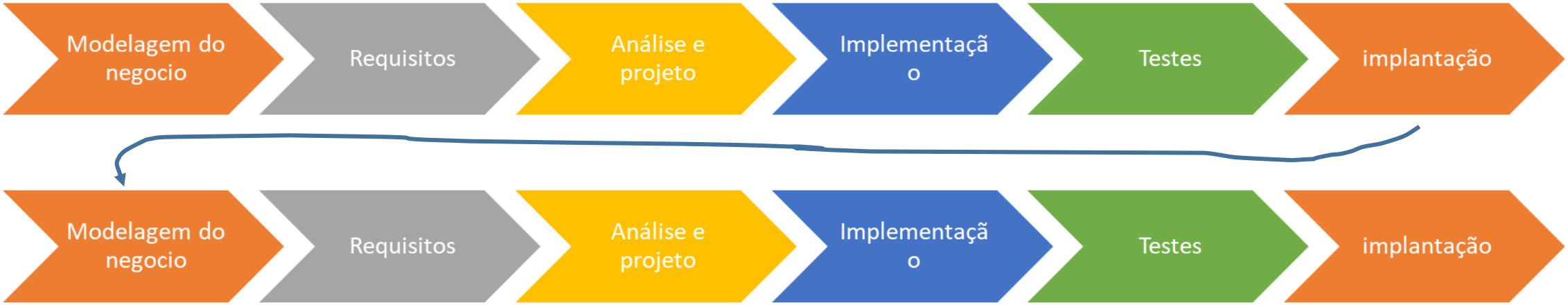
- Isso geralmente resulta em um acúmulo de integração no final da implementação, quando, pela primeira vez, o produto é construído e os testes começam.
- Problemas que permaneceram ocultos durante a Análise, Design e Implementação vêm à tona e o projeto é interrompido quando um longo ciclo de correção de bugs começa.

Problemas e riscos de Sistemas baseados no modelo cascata



Iteração

- Uma maneira mais flexível (e menos arriscada) de proceder é passar diversas vezes pelas diversas disciplinas de desenvolvimento, construindo uma melhor compreensão dos requisitos, projetando uma arquitetura robusta, fortalecendo a organização do desenvolvimento e, eventualmente, entregando uma série de implementações que são gradativamente mais completo.
- Isso é chamado de **ciclo de vida iterativo**.
- Cada passagem pela sequência de disciplinas de processo é chamada de **iteração**.



O que é RUP?

- **R**ational **U**nified **P**rocess
- Metodologia ágil de desenvolvimento de software
- Captura as principais boas práticas modernas de Eng. SW
  - Desenvolvimento de software interativo
  - Gerenciamento de requisitos
  - Uso da arquitetura baseada em componentes
  - Modelagem visual
  - Verificação continua de qualidade
  - Gerenciamento de mudanças
- Tem como objetivo garantir a produção de software de alta qualidade que está de acordo com as necessidades de seus usuários finais com o cronograma e custos previsíveis

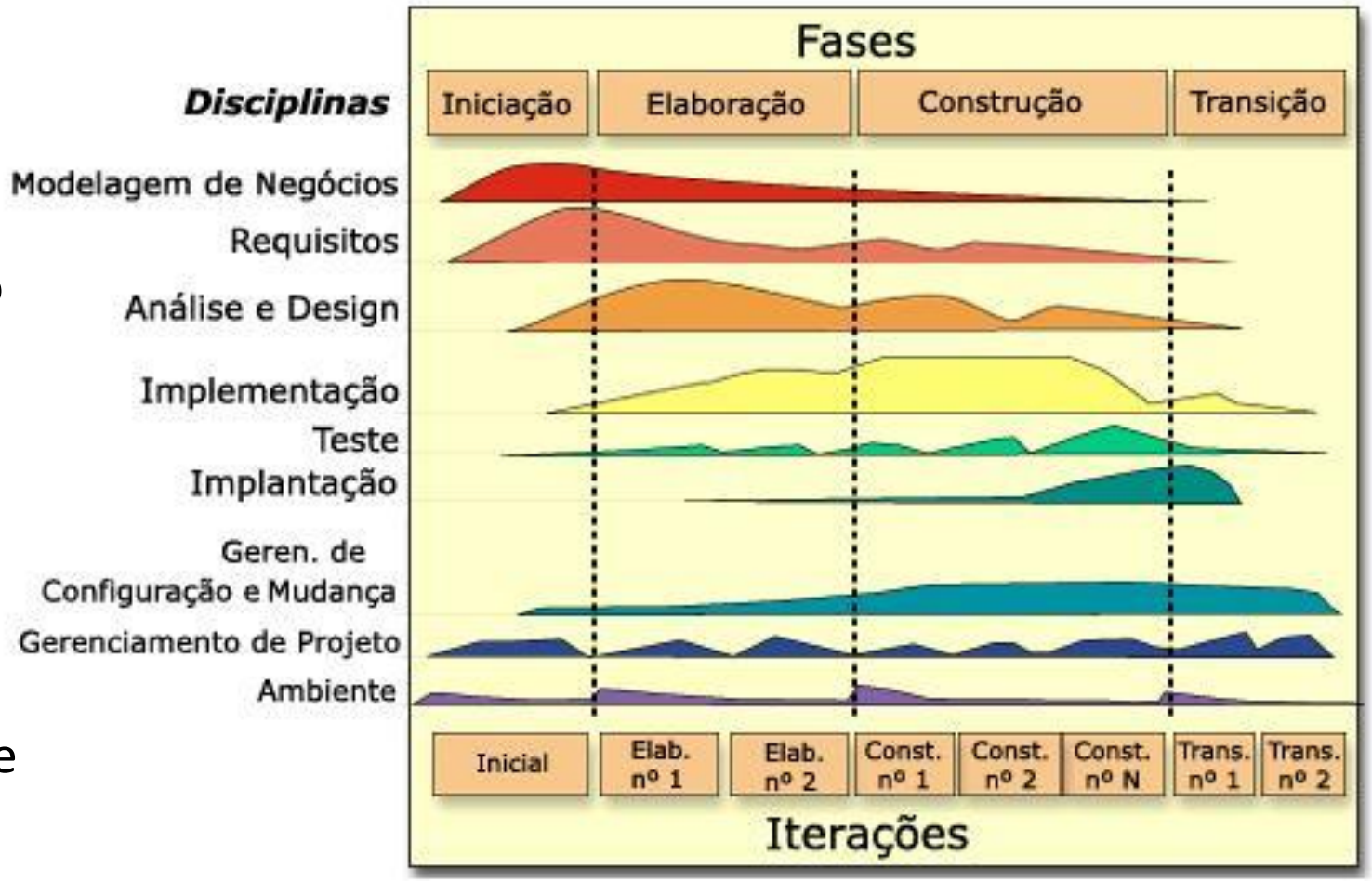
O que é RUP?

- Conjunto de atividades
  - Bem definidas
  - Com responsáveis
  - Com artefatos de entrada e saída
  - Com dependências entre as mesmas e ordem de execução
  - Com modelo de ciclo de vida
  - Descrição sistemática de como devem ser realizadas as atividades
  - UML

Disciplina e fases do RUP

- **Estrutura dinâmica**. A dimensão horizontal representa a estrutura dinâmica ou dimensão temporal do processo. Mostra como o processo, expresso em termos de ciclos, fases, iterações e marcos, se desenvolve ao longo do ciclo de vida de um projeto

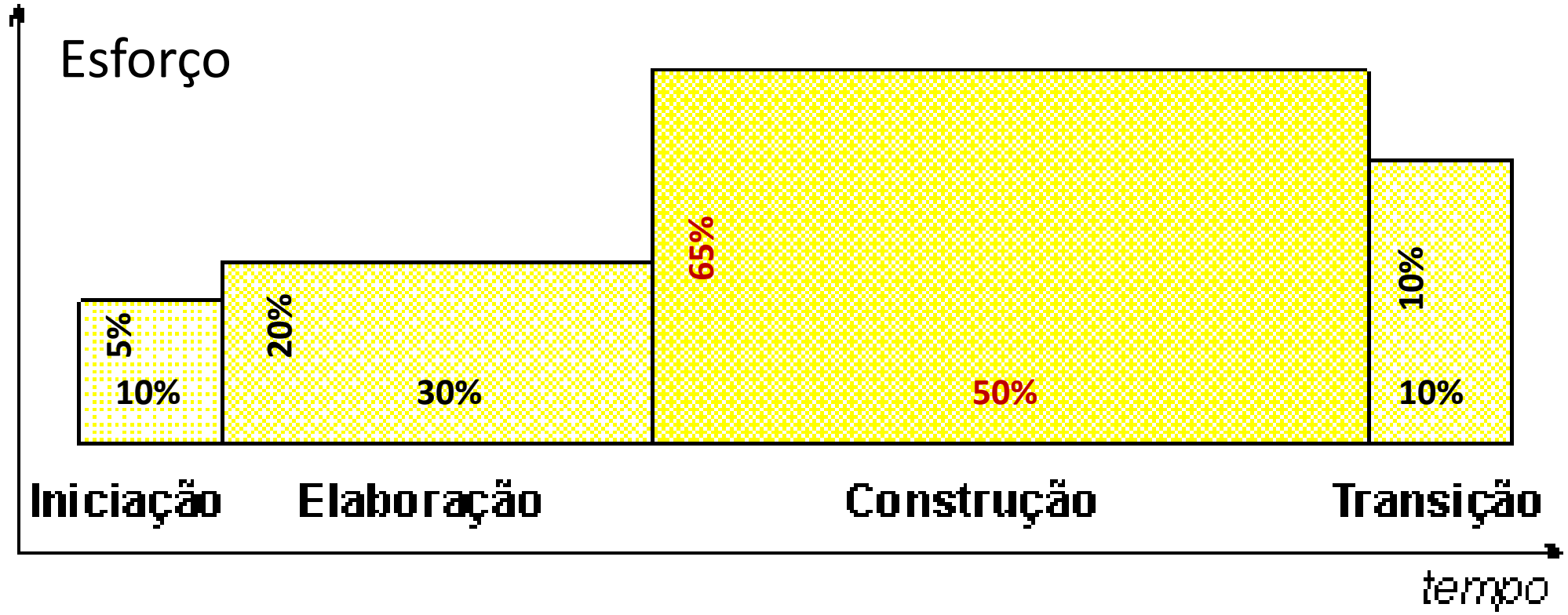
- **Estrutura estática**. A dimensão vertical representa a estrutura estática do processo. Ele descreve como os elementos do processo – atividades, disciplinas, artefatos e funções – são agrupados logicamente em disciplinas centrais do processo (ou fluxos de trabalho).



Fases do RUP

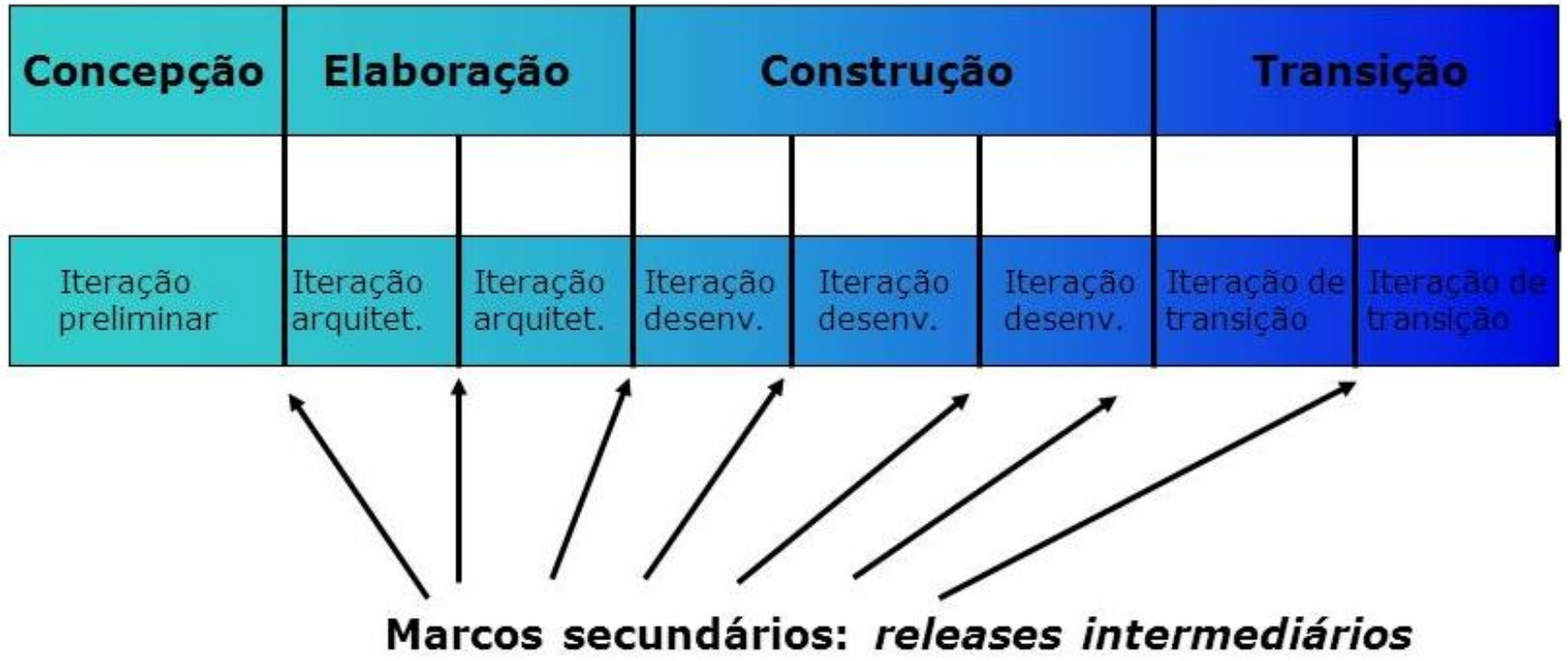

Concepção
Estabelecer o escopo e viabilidade econômica do projeto
Elaboração
Eliminar principais riscos e definir arquitetura estável
Construção
Desenvolver o produto até que ele esteja pronto para beta testes
Transição
Entrar no ambiente do usuário

Cronograma, Esforço e Recursos por Fase



RUP é interativo e incremental

- Cada fase e dividida em iterações
- A ideia é de que ao final de cada iteração eu tenha um incremento no meu software ou componente da arquitetura



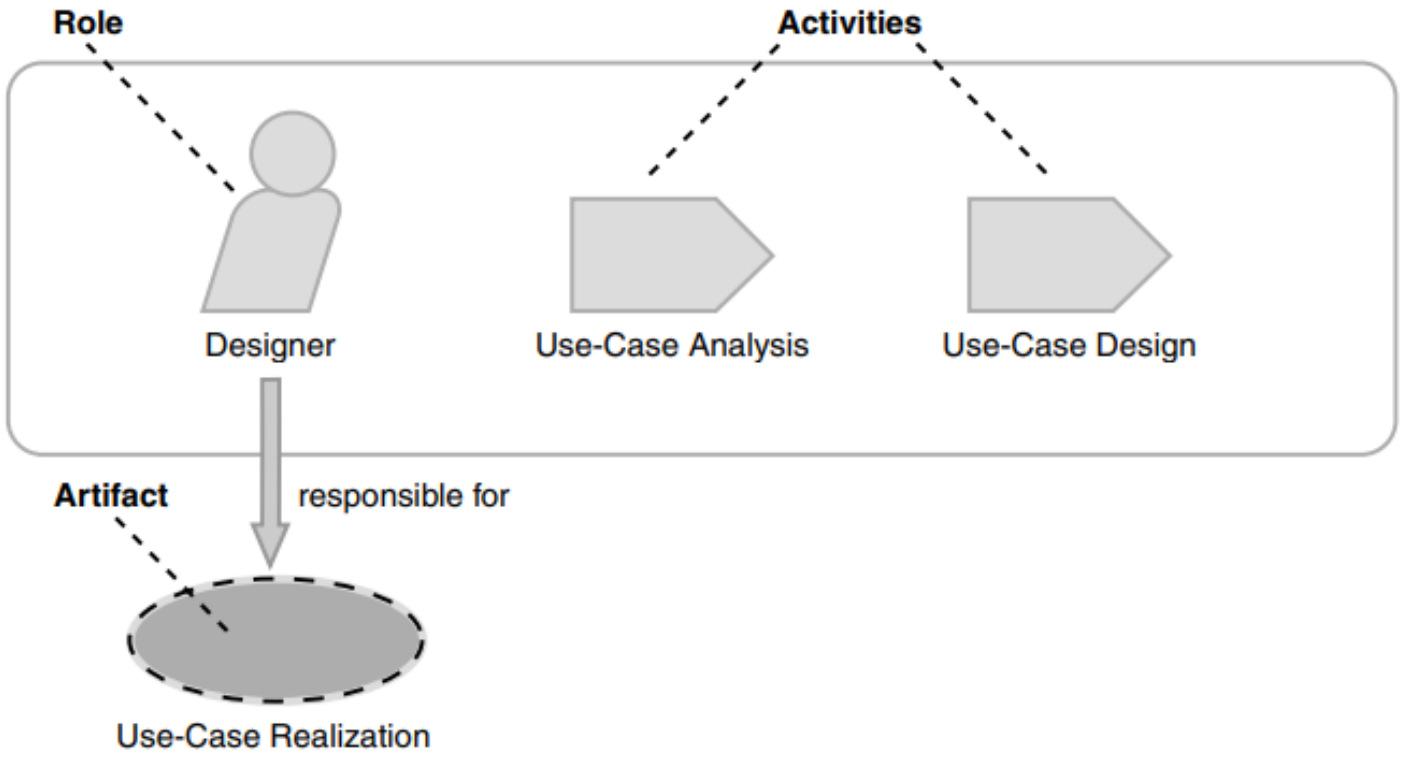
Marcos do projeto

RUP é interativo e incremental

- Cada iteração
  - É planejada
  - Realiza uma sequencia de atividades distintas
  - Resulta em uma versão executável do sistema
  - É avaliado segundo critérios de sucesso previamente acordados

Quatro Elementos Chave de Modelagem do RUP

- **Papeis ou Funções**. “Quem”
- **Atividades**. “Como”
- **Artefatos**. “O quê”
- **Fluxos de trabalho**. “Quando

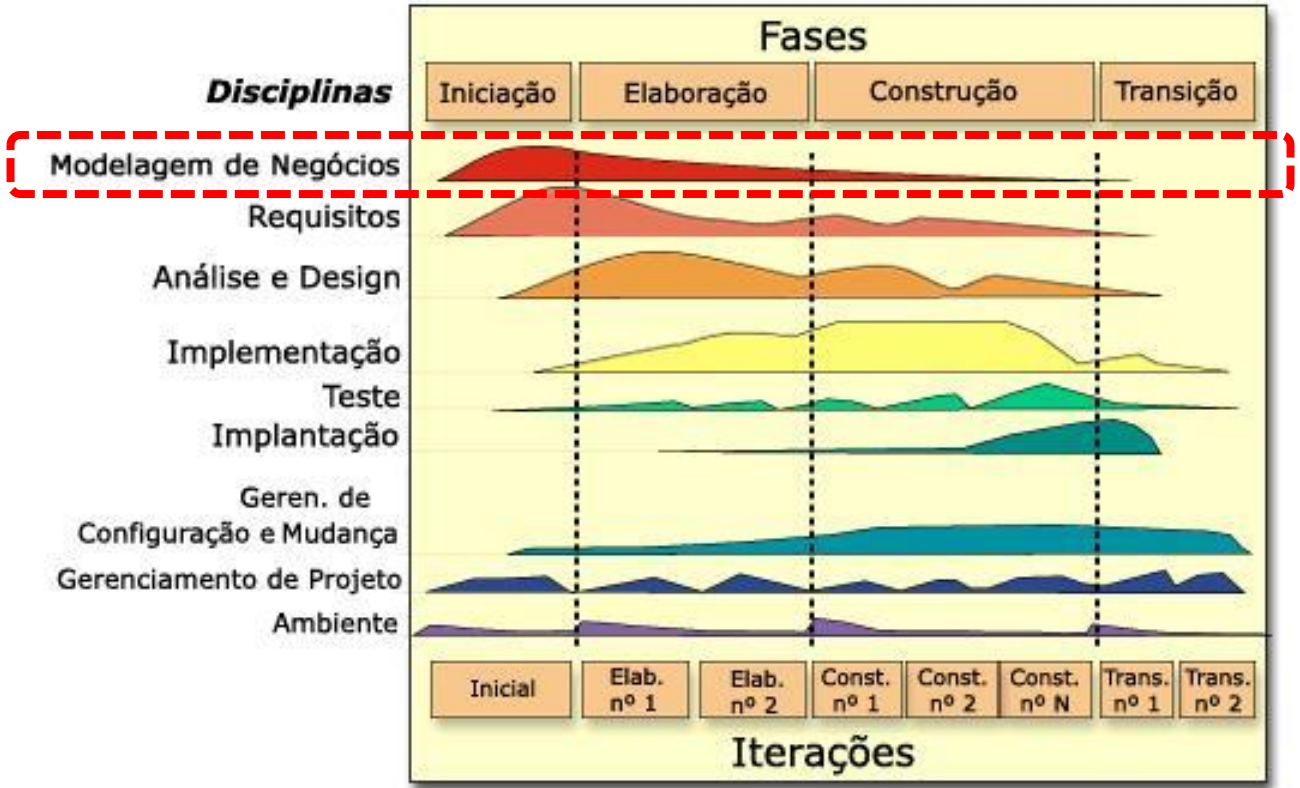


Disciplinas do RUP

- Modelagem de Negocio
- Requisitos
- Análise e projeto
- Implementação
- Testes
- Implantação
- Gerenciamento e Planejamento
- Gerenciamento de configuração e mudanças
- Ambiente

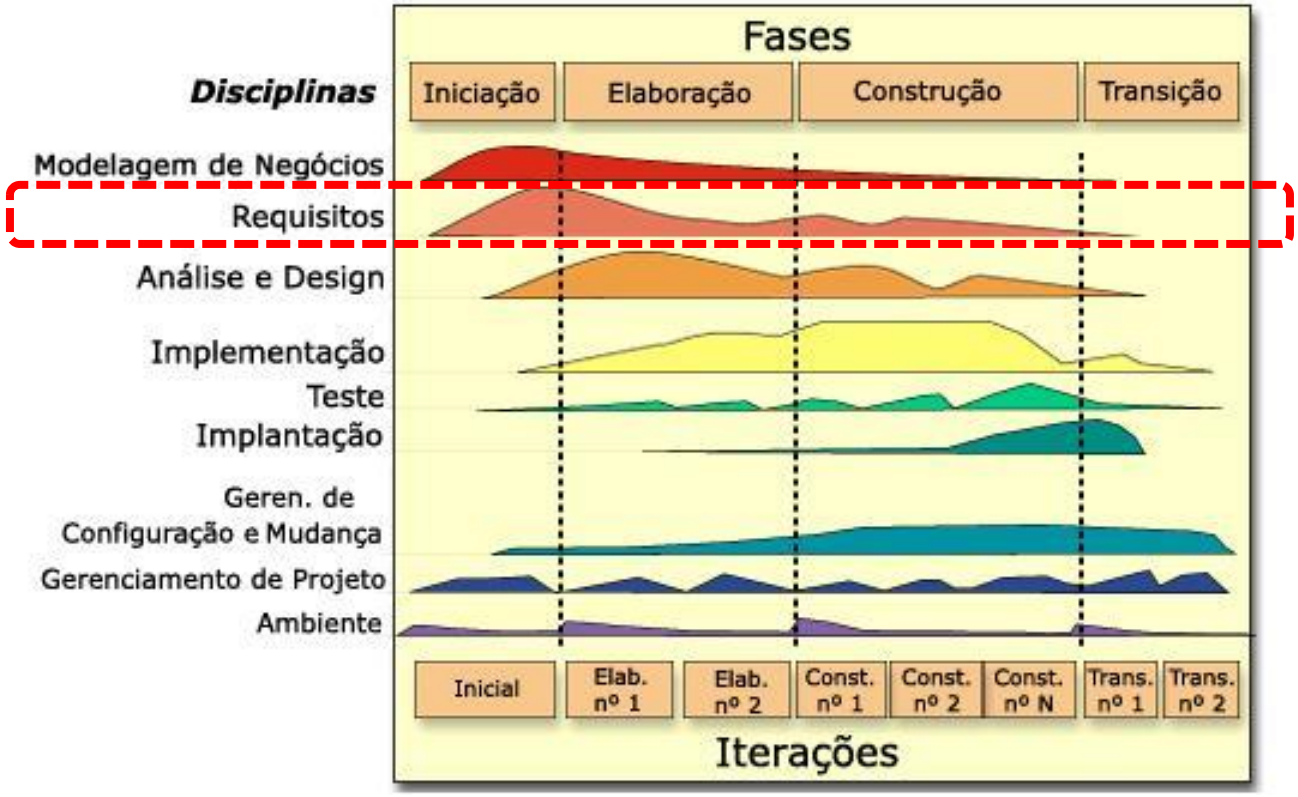
Modelagem de Negócio

- Entender a estrutura dinâmica da Organização
- Entender os problemas e identificar as melhorias em potencial



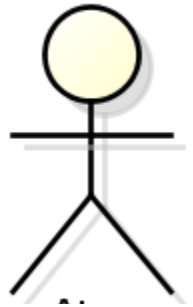
Requisitos

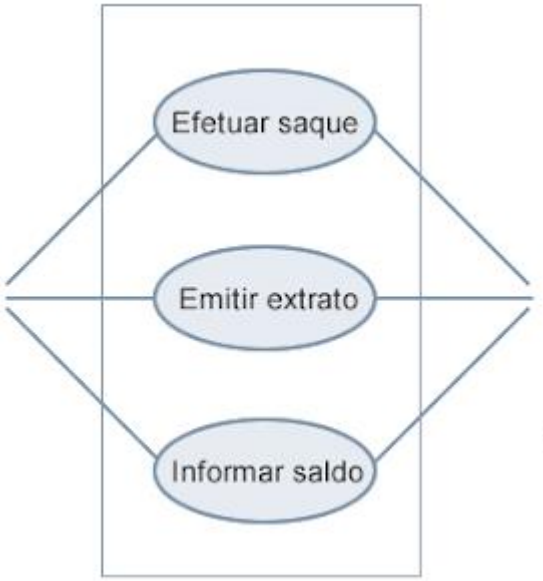
- Estabelecer e manter a concordância entre o cliente e "stakeholder" sobre o que o sistema irá fazer
- Definir os limites do sistema
- Prover uma base para estimar tempo e ciclo de desenvolvimento



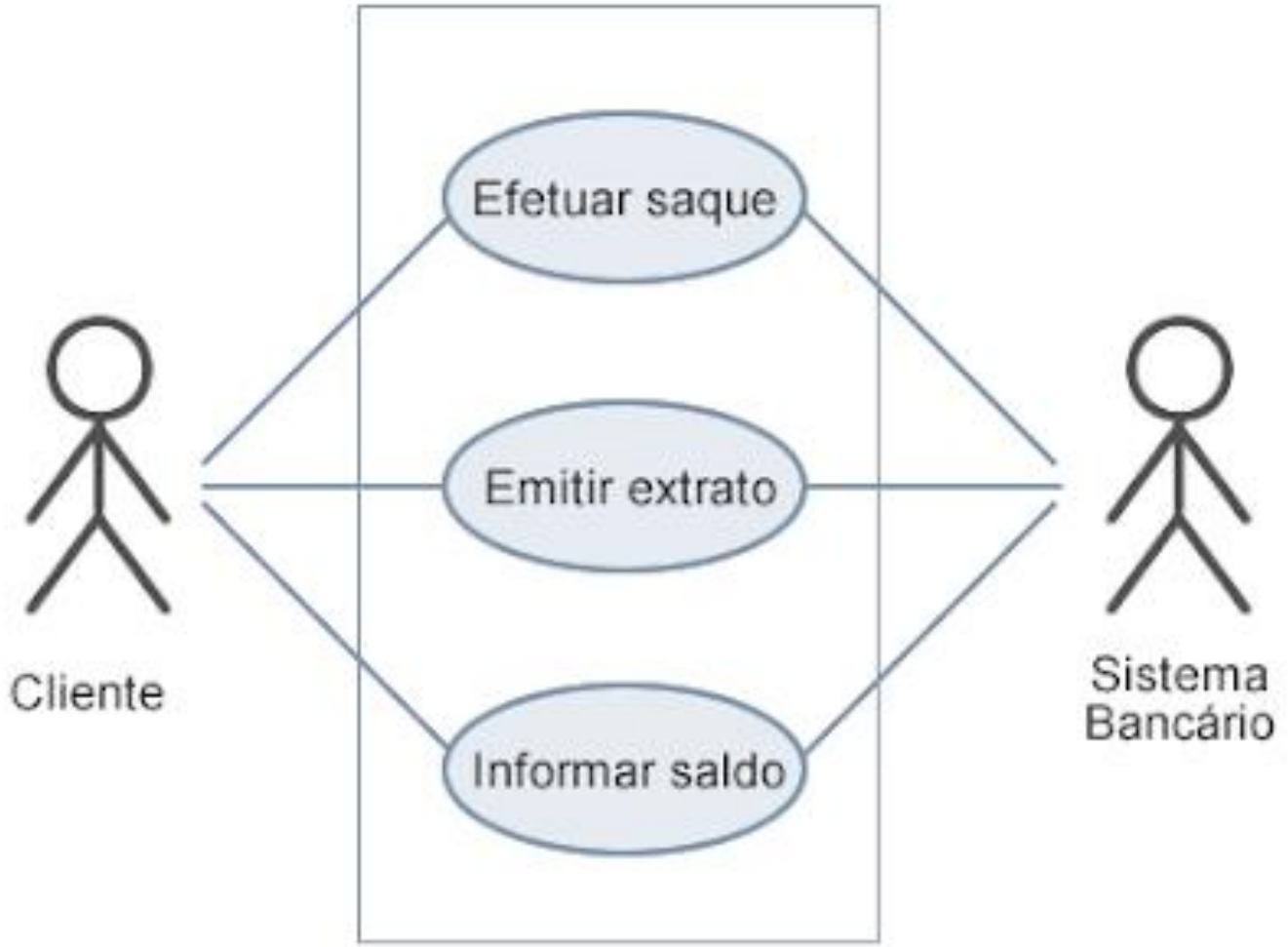
Conceitos

- No RUP a fase de requisitos é fortemente ligada a UML:
- Ator
  - Categoriza as entidades que interagem com o sistema
  - Externas ao sistemas
  - Pode ser usuários ou outros sistemas
- Caso de uso
  - Sequencia d ações que o sistema executa para produz um resultado visível para um ator
  - “Uma forma de usar o sistema”





Exemplo de Diagrama de Caso de Uso



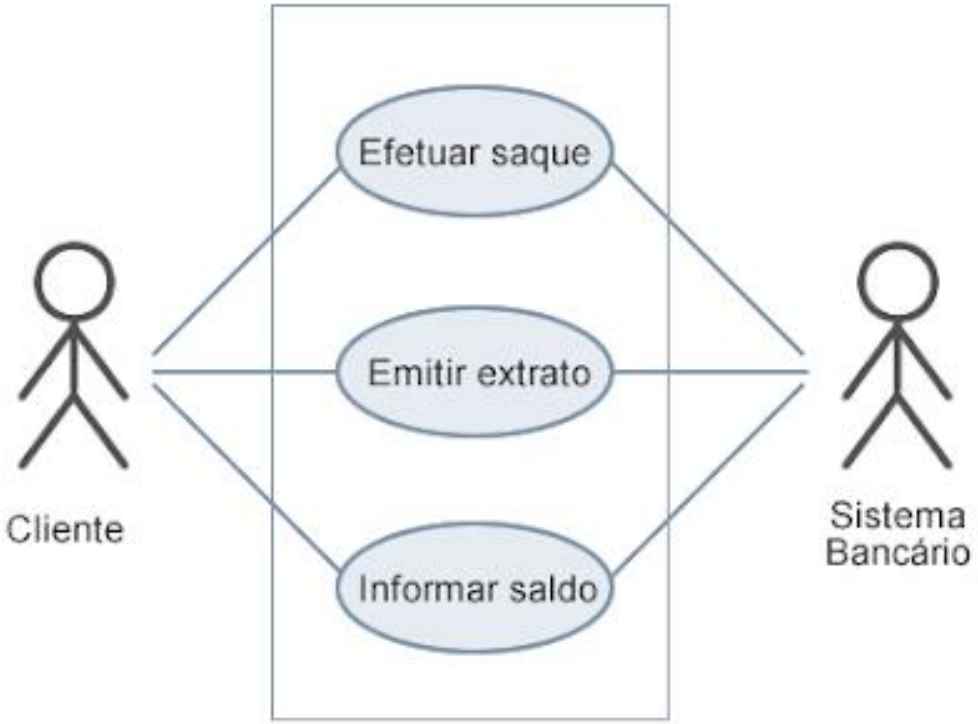
Detalhamento de Casos de Uso

- Nome
- Descrição
- Fluxo e eventos
- Diagrama de estados e atividades
- Pré-condições e pós–condições
- Casos de uso relacionados
- Requisitos não funcionais relacionados

Fluxo de eventos

- Um fluxo básico ou “caminho Feliz”

- Zero ou mais fluxos alternativos
  - Variações
  - Erros



Exemplo de Fluxo de Eventos
Caso de uso: Sacar Dinheiro

Normal ("caminho feliz")

1. Inserir cartão
2. Selecionar "SAQUE"
3. Informar valor
4. Informar Senha
5. Retirar Dinheiro

Fluxos alternativos

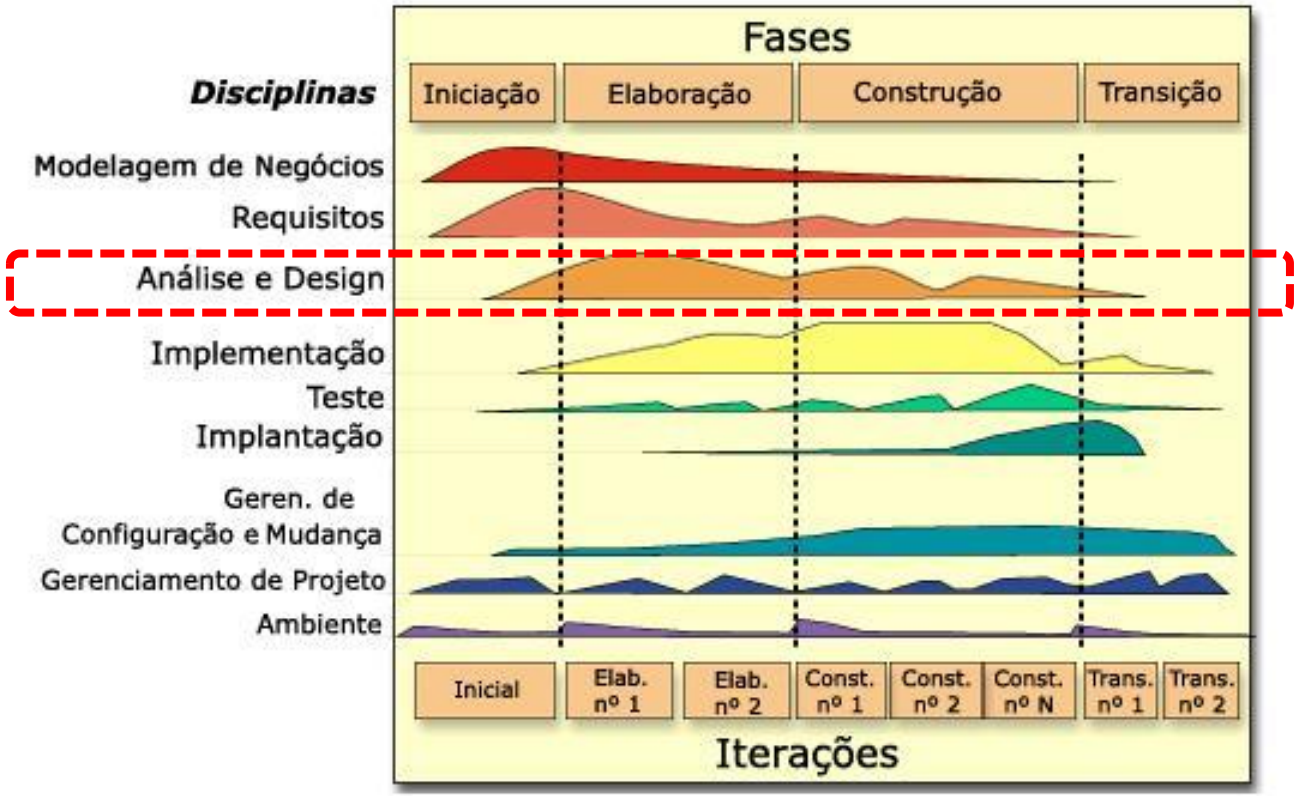
- Sem Saldo
  - Após o passo 4 informar que não há saldo suficiente
- Senha inválida
  - Após o passo 4 informar a senha não é válida
  - Repetir o passo 4 novamente (no máximo 3 vezes)

Outros Arterfatos

- Especificação suplementar
  - Requisitos não funcionais
    - Ex: Tem que ter desempenho, fácil uso
  - Lista funcionalidades comuns a vários casos de uso
    - Este caso de uso vai ter login, o outro não
- Glossário
  - Lista de termos comuns o projeto
  - Facilita a comunicação entre o time
    - Ex: o que é matricula do aluno? Conjunto formado pelo ano de ingresso, mais o código da faculdade, mais código curso mais sua colocação

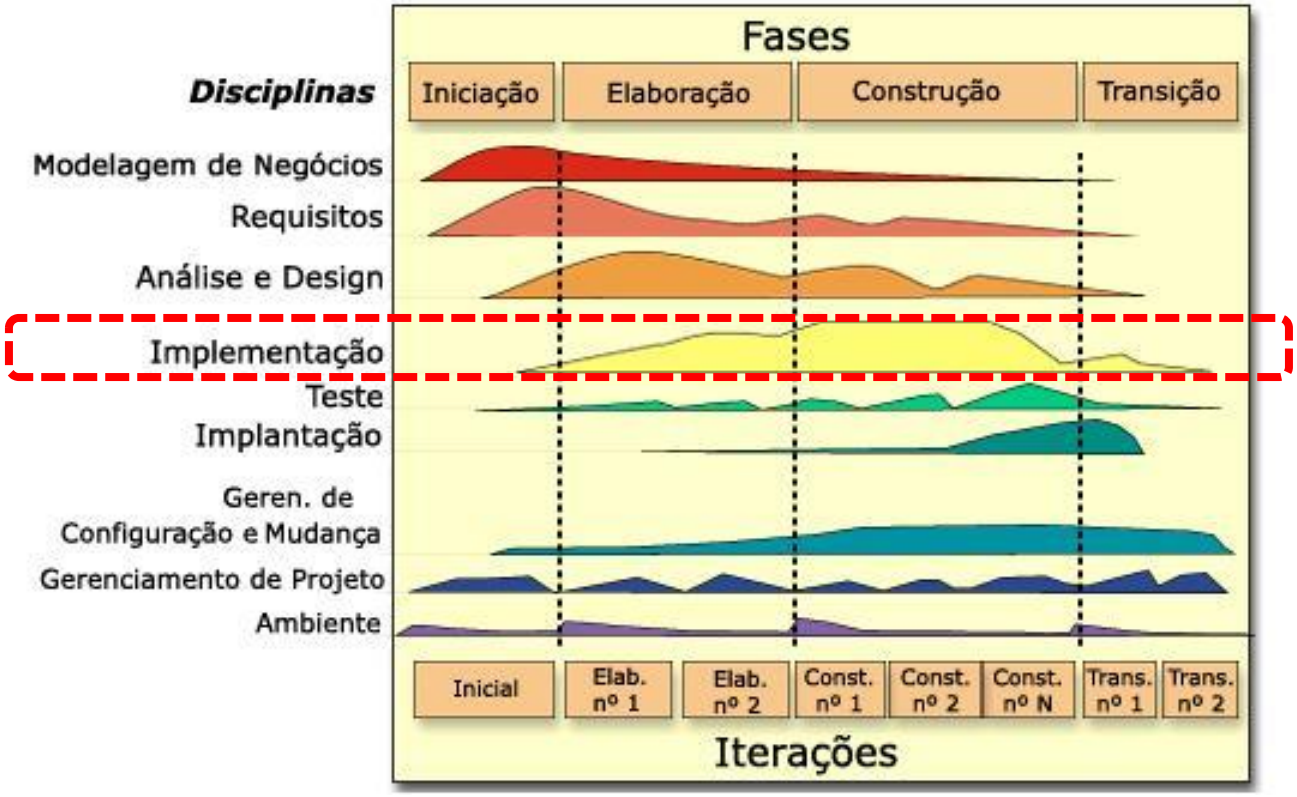
Análise e Projeto

- Transformar os requisitos em um projeto do sistema
- Construir uma arquitetura robusta para o sistema
- Adaptar o projeto para as limitações do ambiente de execução



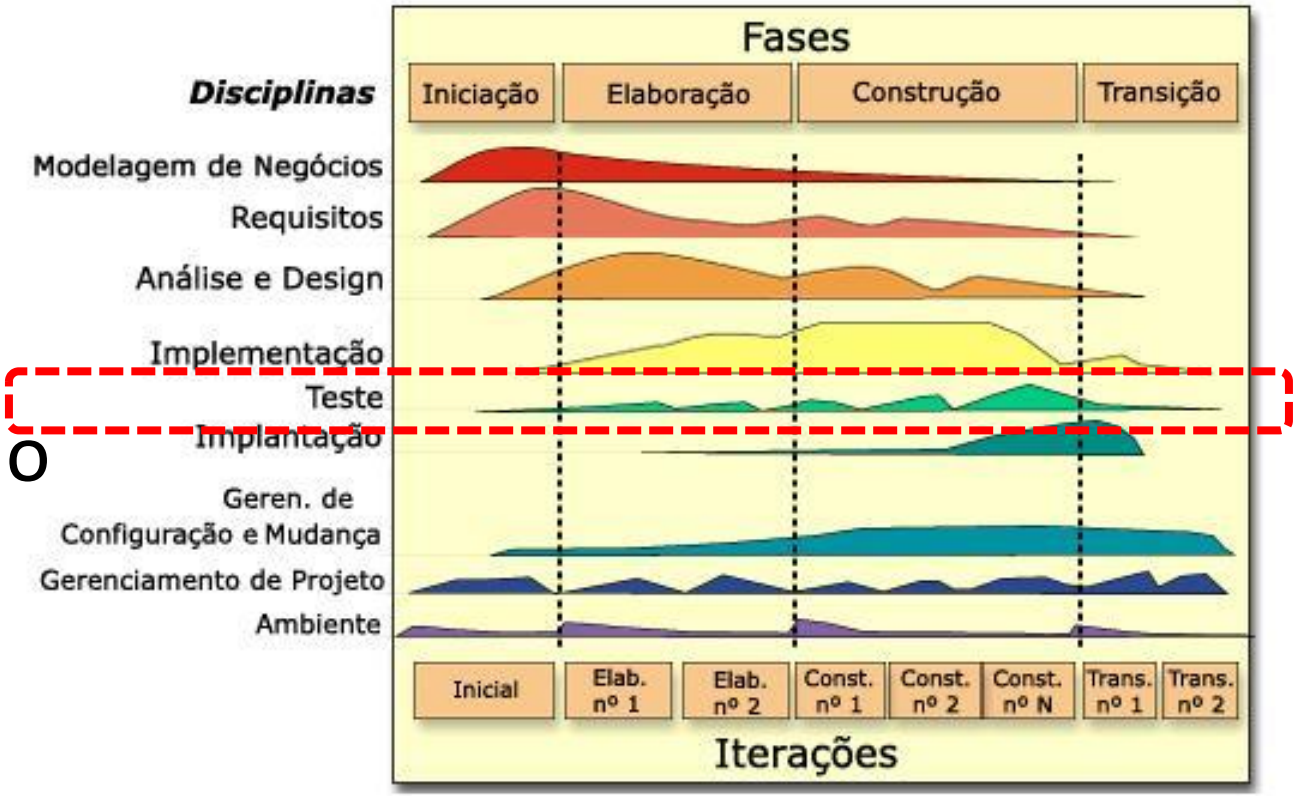
Implementação

- Definir e organizar o código
- Implementar classes e objetos
- Testar unidades
- Integrar unidades



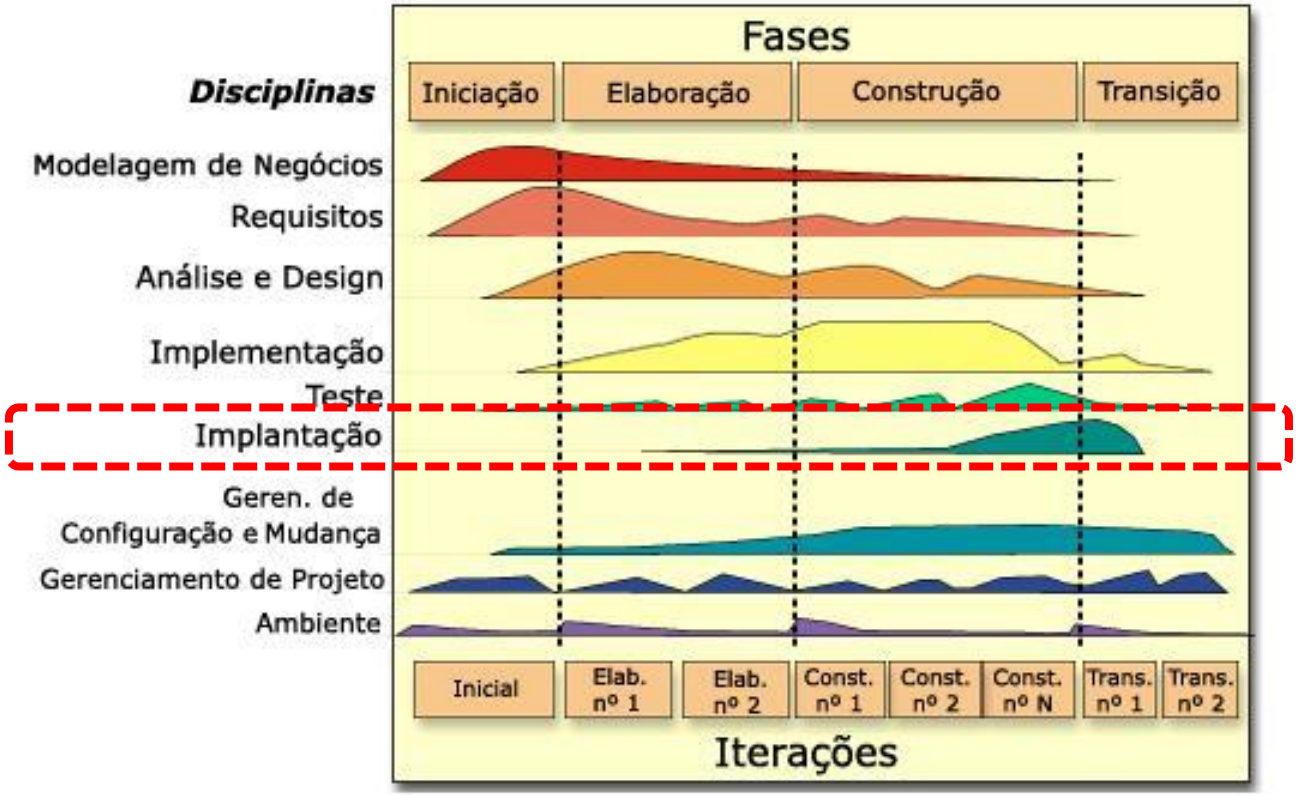
Teste

- Encontrar e documentar defeitos
- Validar sistemas atende ao especificado
- Validar se o sistemas construído como projetado



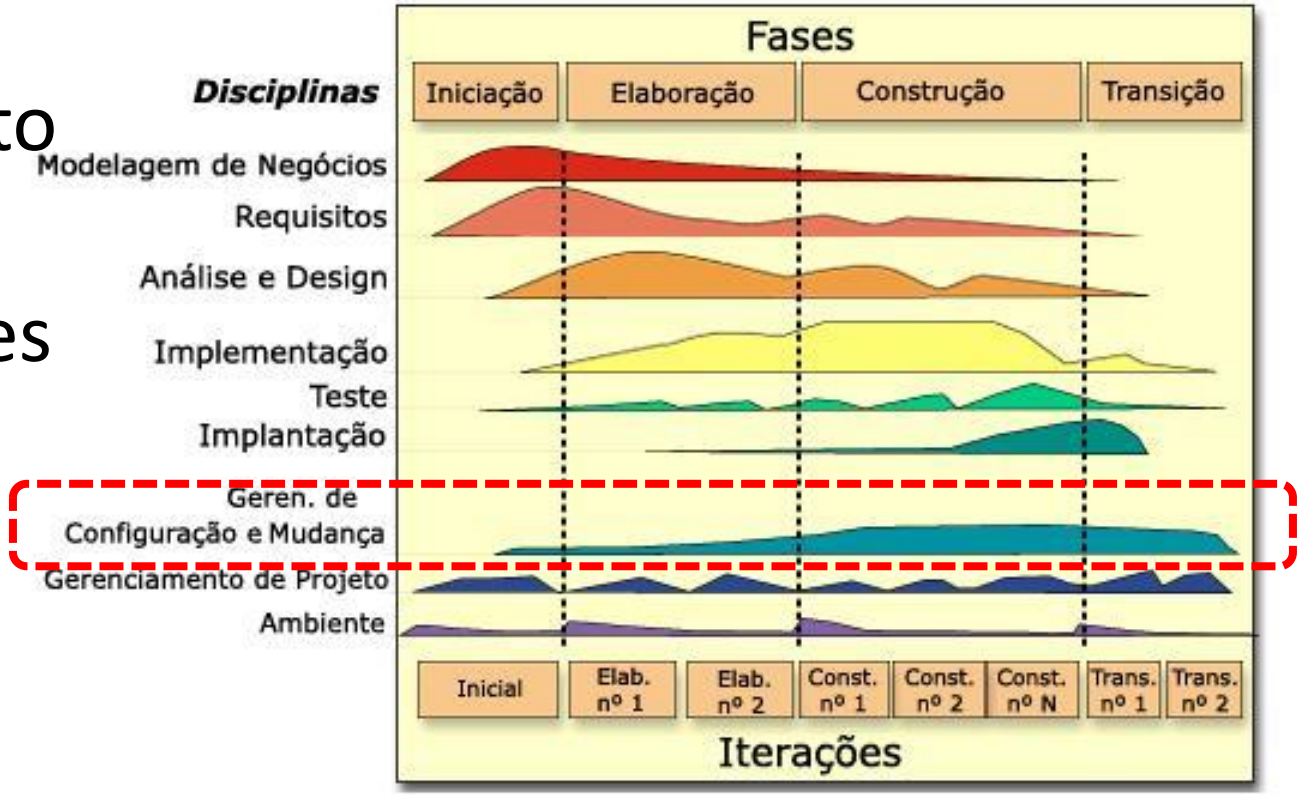
Implantação

- Garante que o sistema está disponível para o usuário final



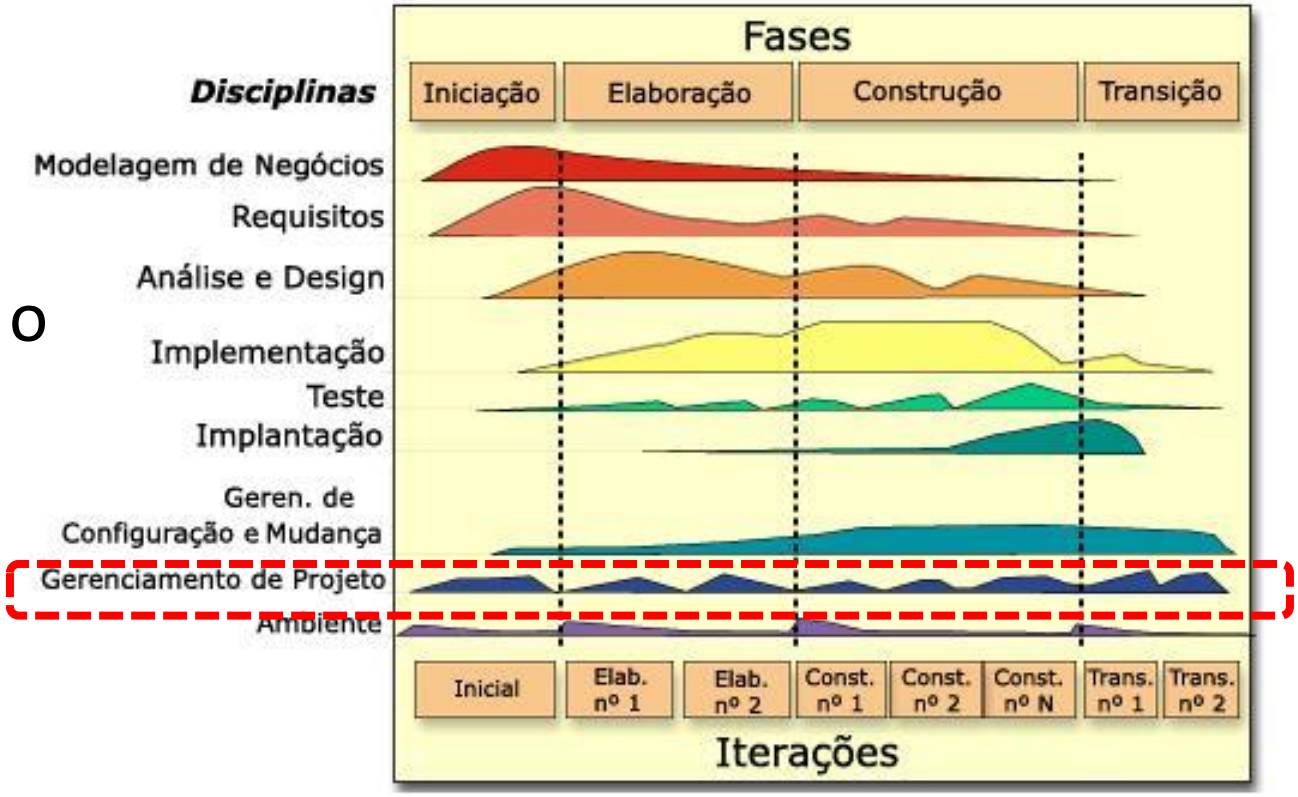
Disciplinas de Apoio
Gerência de Configuração e Mudanças

- Controlar os artefatos produzidos no desenvolvimento do projeto
- Evita a ocorrência dos seguintes problemas
  - Atualizações simultâneas
  - Múltiplas versões
  - Notificação limitada



Disciplinas de Apoio Gerenciamento de Projeto

- Framework para gerência do projeto
- Disponibilizar guias para planejar, executar, acompanhar e monitorar o projeto
- Gerenciamento de riscos



Ambiente

- Focado nas atividades relacionadas ao processo
  - Processo organizacional -> processo do projeto
  - Refinamento do processo do projeto