

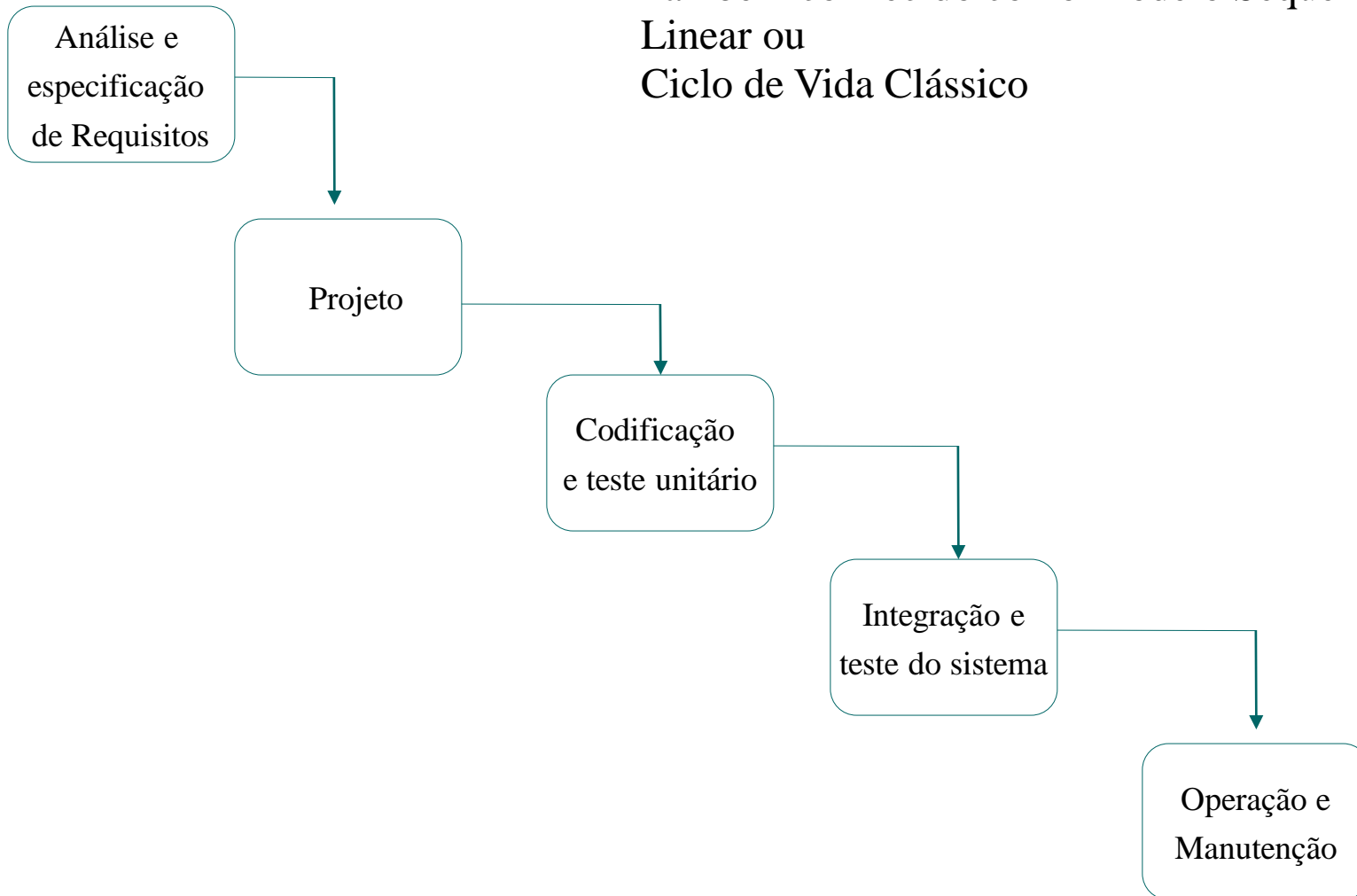
Universidade de Caxias do Sul

Paradigmas de Processo

Profa. Iraci Cristina da Silveira De Carli
icsilvei@ucs.br

Modelo em Cascata

Também conhecido como Modelo Seqüencial Linear ou
Ciclo de Vida Clássico



Sommerville

Modelo em Cascata

➤ Aplicação

- Casos em que os requisitos de um problema são bem compreendidos e estáveis

➤ Características

- Fases realizadas sequencialmente
- Cada fase gera um documento que deve ser aprovado antes de seguir para fase seguinte
- Resultado de uma fase é entrada da outra
- Implementação do produto deve ser postergada até que os objetivos tenham sido entendidos completamente

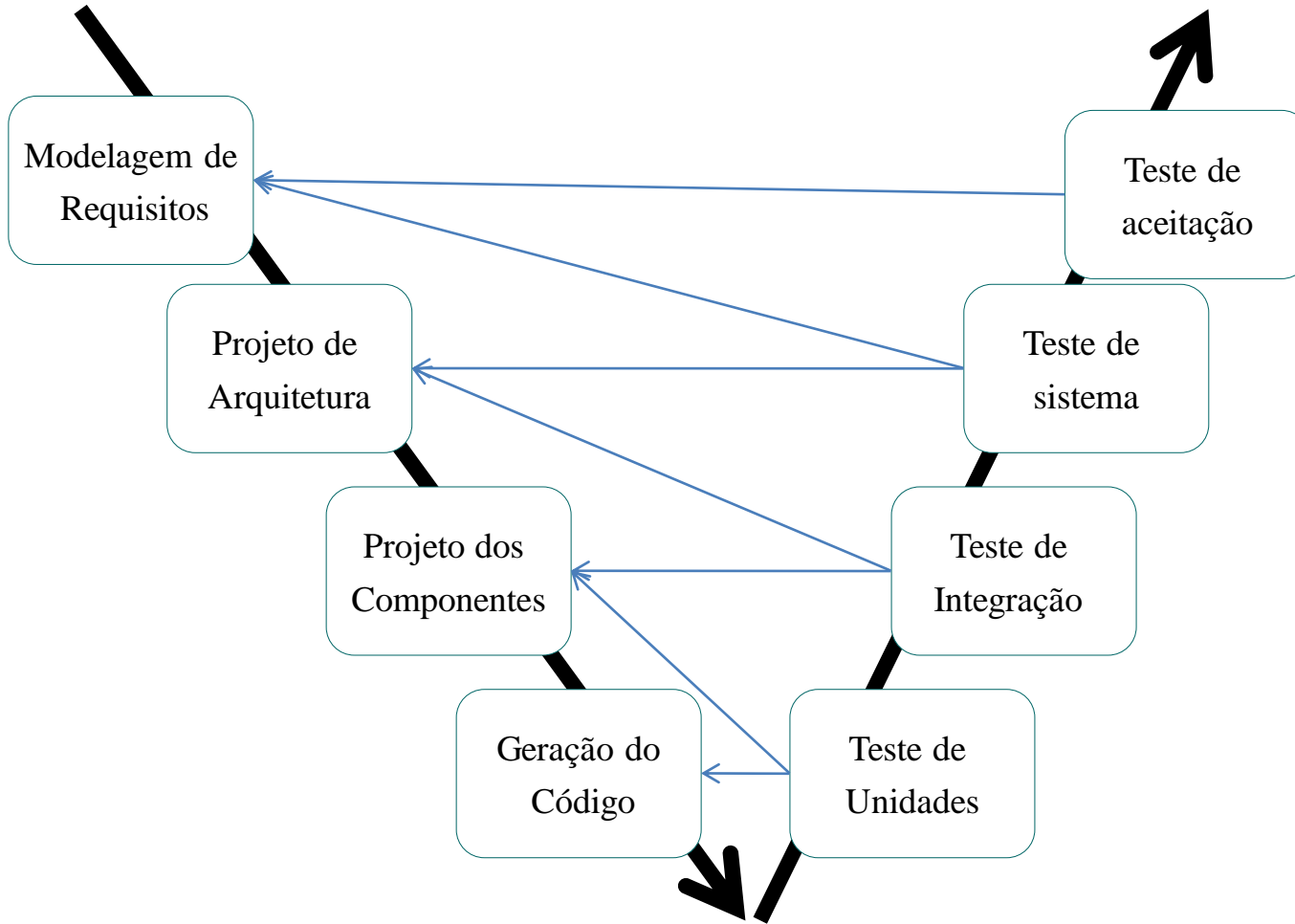
➤ Benefícios

- Simples gerenciamento
- Primeiro processo de produção
- Bastante usado – melhor que não usar nada

➤ Problemas

- Software disponibilizado depois de muito tempo
- Fluxo sequencial não corresponde a realidade
- Necessita compreensão completa dos requisitos necessários de forma antecipada, porém aprende-se durante todo processo
- Estado de bloqueio – equipe aguarda atividades anteriores serem concluídas

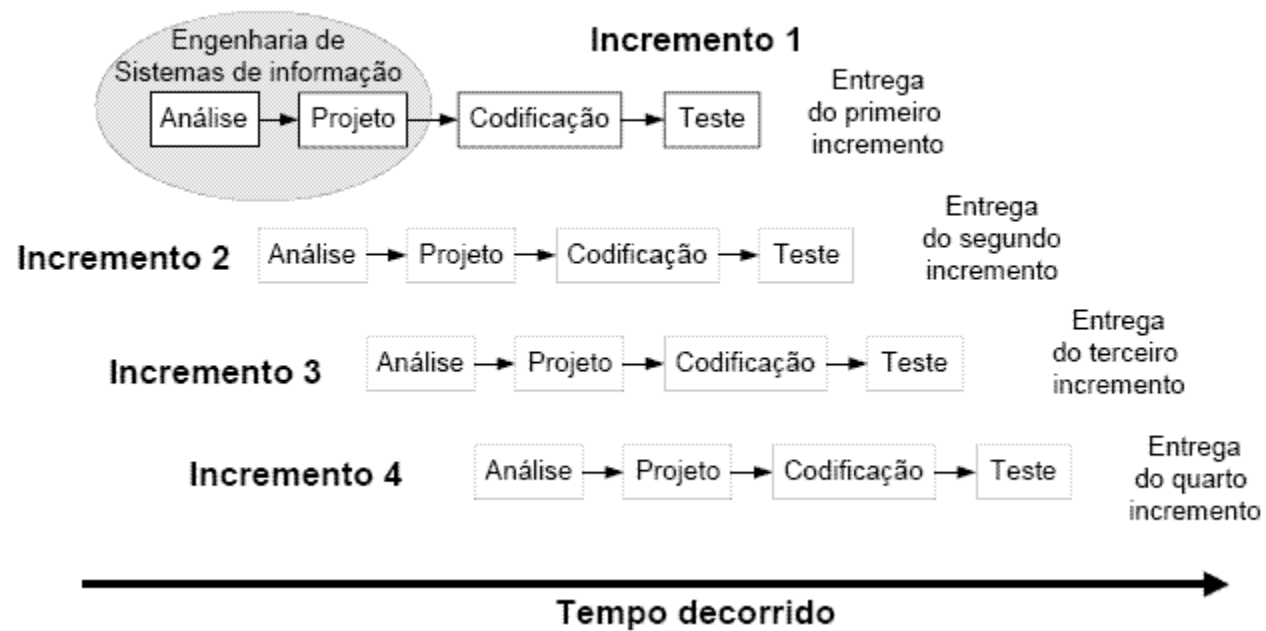
Modelo em V



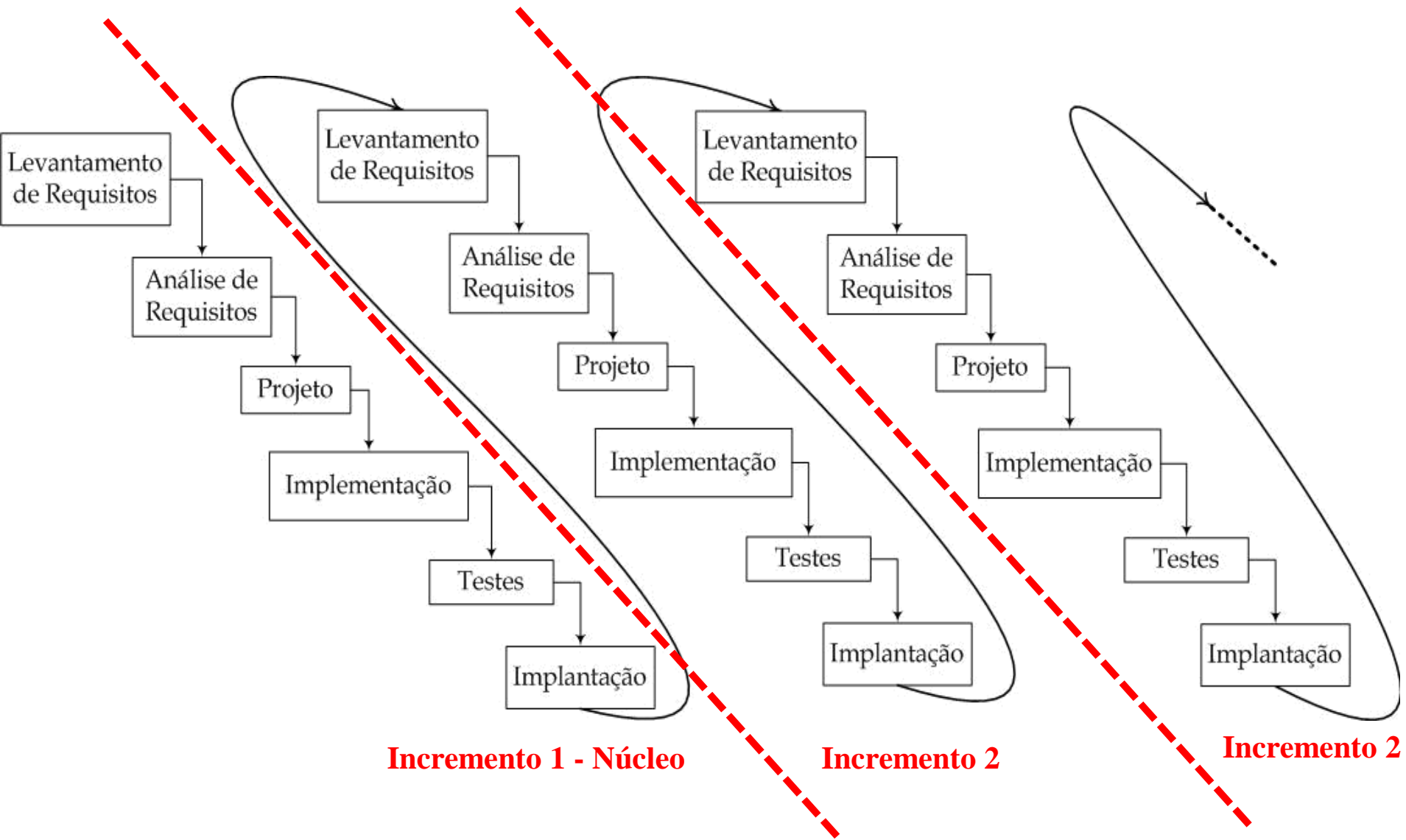
Modelo em V

- Características
 - Variação do modelo cascata
 - Representa a relação entre ações de garantia de qualidade e as ações de construção do software

Modelo Incremental



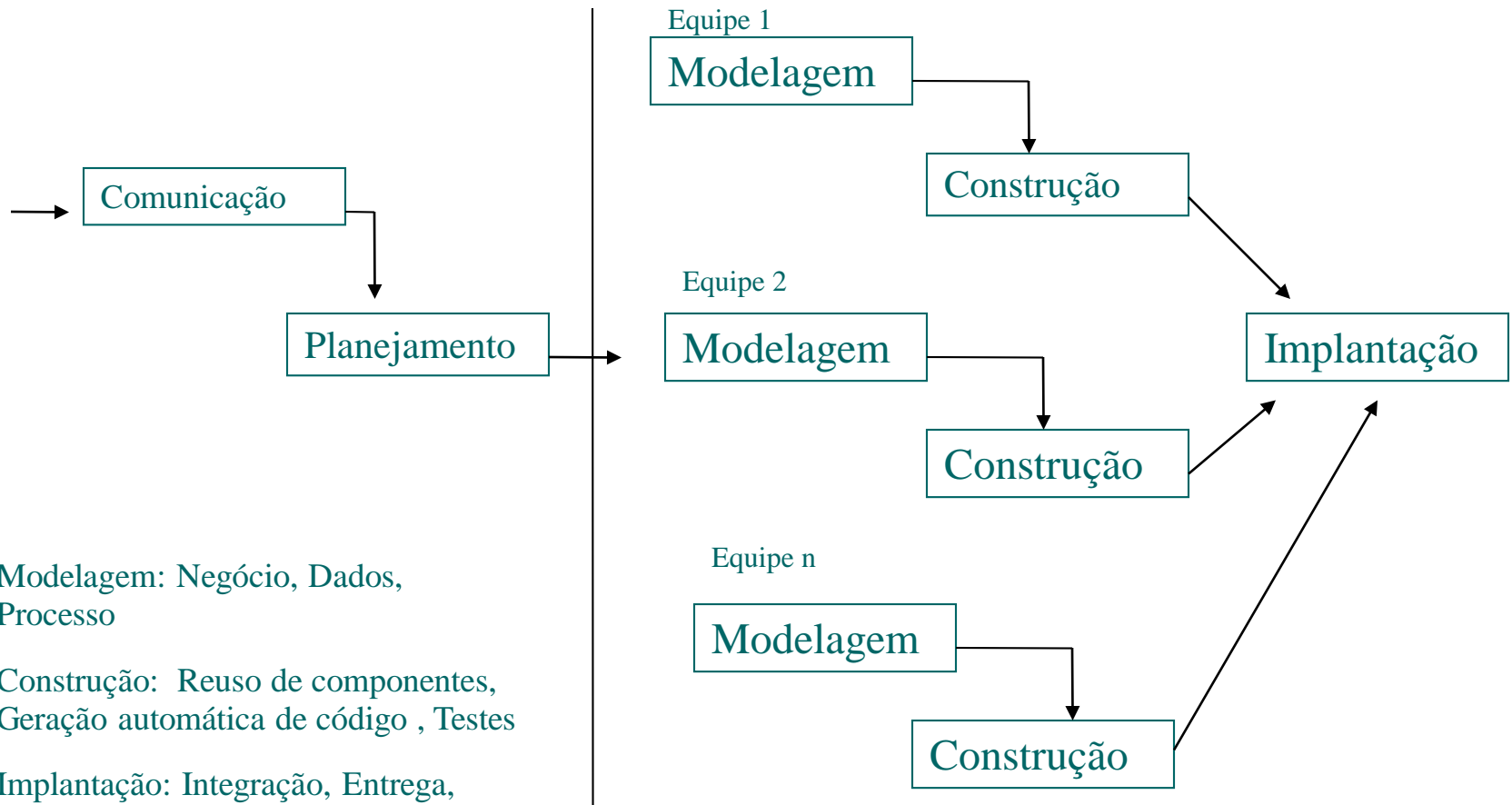
Vários cascatas



Modelo Incremental

- **Características**
 - Necessidade de entrega rápida de um conjunto de funcionalidades ao usuário
 - Combina o cascata com a filosofia interativa
 - Aplica seqüência linear das fases várias vezes
 - Cliente adquire experiência
- **Benefícios**
 - Cliente roda o sistema antes com as funcionalidades mais importantes
 - Proporcionar aos clientes algumas oportunidades de adiar decisões sobre seus requisitos detalhados
 - A utilização de um incremento facilita a definição dos requisitos dos próximos
 - Reduzir o retrabalho no processo de desenvolvimento
 - O risco de fracasso é menor
- **Problemas**
 - Tamanho dos incrementos
 - Incrementos devem ser pequenos 20 mil linhas e produzir alguma funcionalidade para o sistema.
 - Pode ser difícil mapear os requisitos do usuário nesse tamanho de incremento
 - As funções em comuns devem estar disponíveis antes e alguma função básica pode ficar mal definida

Modelo RAD (Rapid Application Development)



Modelagem: Negócio, Dados, Processo

Construção: Reuso de componentes, Geração automática de código, Testes

Implantação: Integração, Entrega, Feedback

Modelo RAD

- Rapid Application Development
- Enfatiza ciclo de desenvolvimento curto
- Adaptação do modelo cascata em alta velocidade
- Abordagem de construção baseada em componentes
- Geração automática de código
- Várias equipes trabalhando juntas

Modelo de Prototipagem

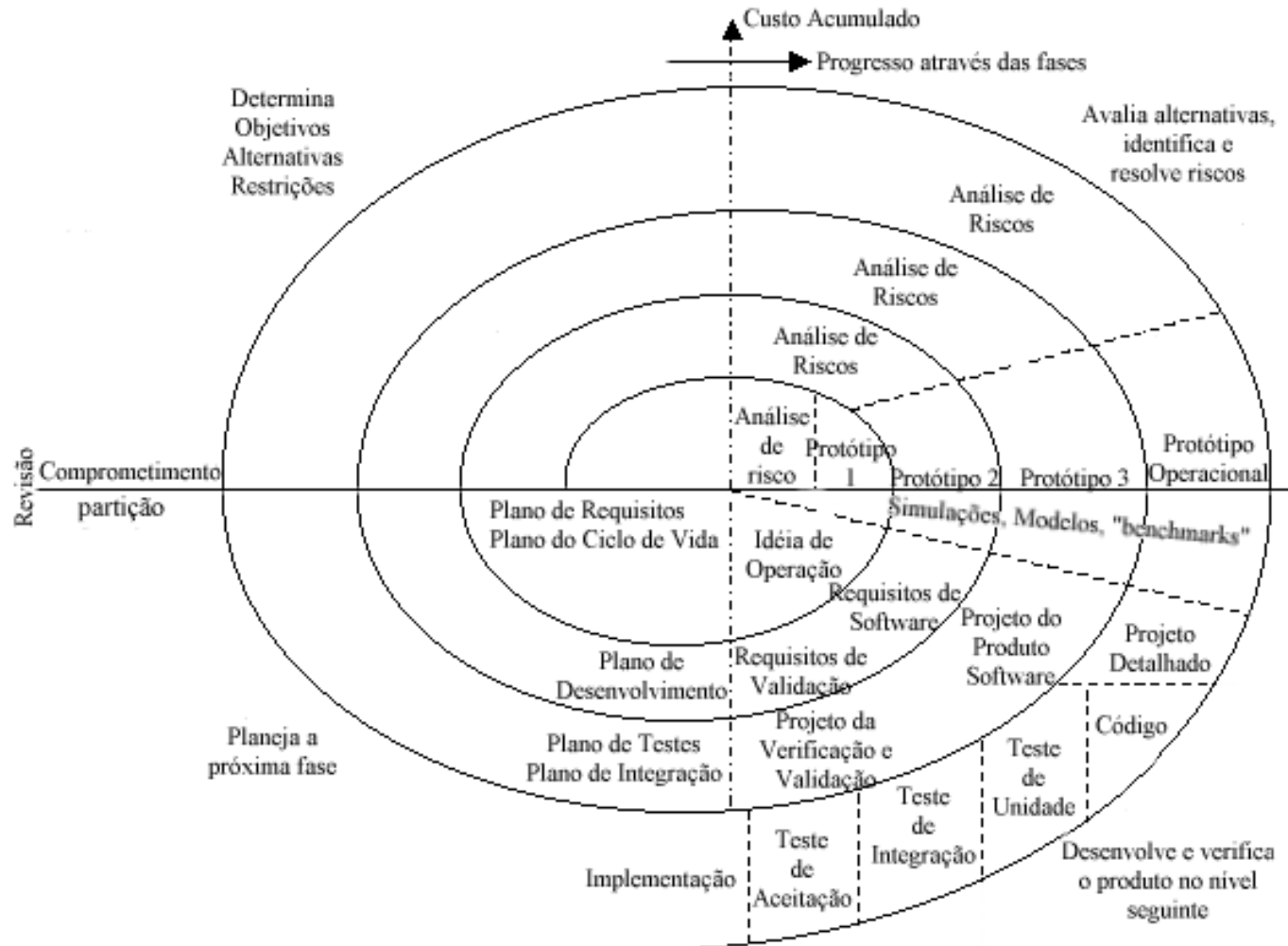


Pressmann

Modelo de Prototipagem

- Características
 - Ajudar o cliente a compreender as necessidades na fase de análise e especificação de requisitos
 - Quando existe dúvida na eficiência de um algoritmo, adaptabilidade de um sisop ou na interação homem-máquina
 - Usado na etapa de definição de requisitos junto a outros modelos
- Tipos
 - Descartável
 - Transforma em produto

Modelo Espiral



Pressmann

Modelo Espiral

- Características
 - Reconhecimento explícito de risco
 - Combina a natureza interativa da prototipagem com os aspectos sistemáticos do modelo seqüencial linear
 - No lugar de representar atividades com retorno para outra utiliza uma espiral
 - Fornece potencial para o desenvolvimento rápido de versões incrementais
- Cada loop na espiral representa uma fase processo de software: viabilidade do sistema, definição de requisitos, projeto...
- Benefícios
 - Abordagem realística para sistemas de grande porte
 - Desenvolvedor e cliente entendem bem e reagem mais rápido aos risco.
 - Reduzir os riscos antes de virarem problemas
- Problemas
 - Convencer cliente que paradigma é controlável
 - Competência na avaliação de riscos

Modelo Espiral

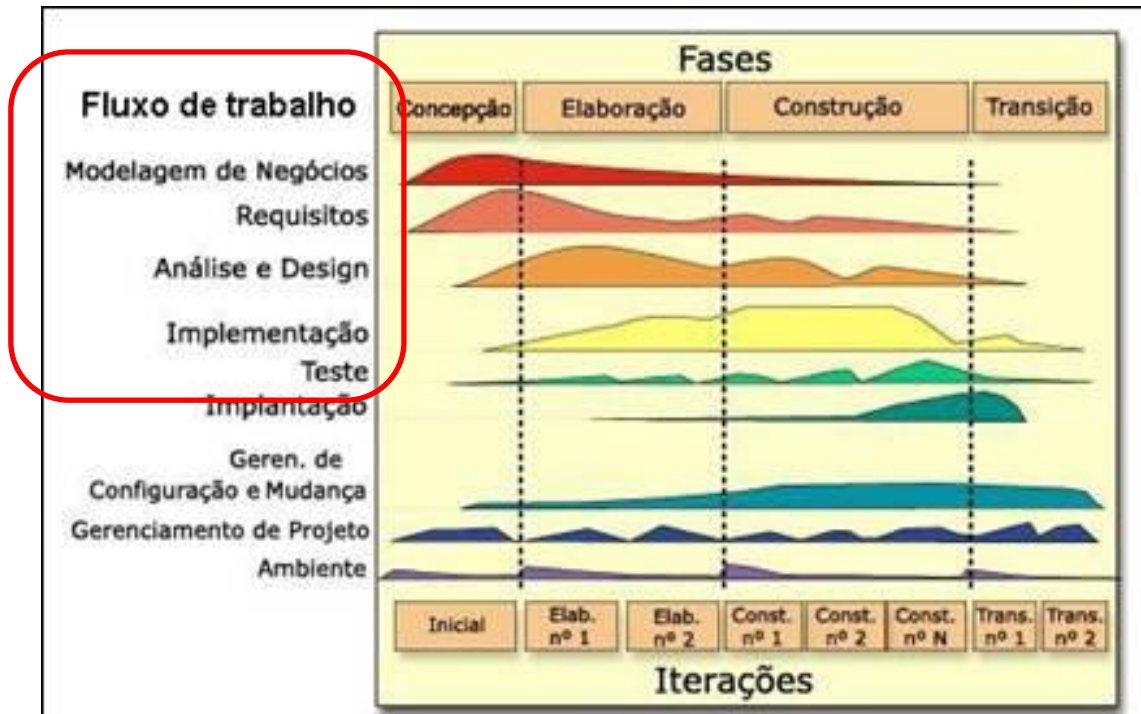
- Setores do Loop
 - Definição de objetivos
 - Identificar restrições de processo e produto
 - preparar um plano de gerenciamento
 - Identificar os riscos do projeto
 - Avaliação e redução de riscos
 - analisar detalhadamente
 - tomar providências necessárias para reduzir os riscos
 - exemplo: risco de requisitos inadequados – fazer um protótipo
 - Desenvolvimento e validação
 - Escolher modelo de desenvolvimento
 - riscos relacionados a interface- prototipação
 - riscos de segurança - transformações formais
 - integração de sistemas - cascata
 - Planejamento
 - projeto é revisto e é tomada a decisão de continuar com a próximo loop.

Características do Processo Unificado

- Estrutura estática
- Estrutura dinâmica
- Iteração

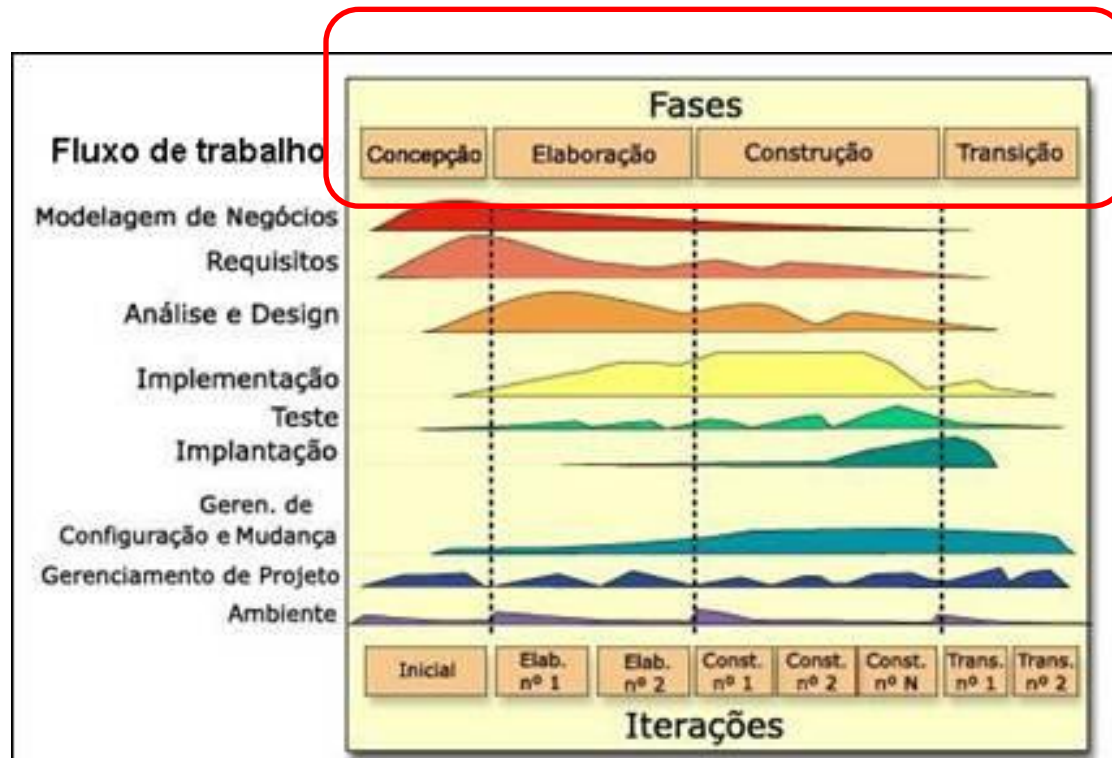
Características do Processo Unificado

- Estrutura estática
 - Cada workflow é um conjunto de atividades
 - Atividade é uma tarefa executada por um trabalhador a fim de produzir um artefato
 - Um trabalhador é um papel que um indivíduo pode desempenhar no projeto
 - Um artefato é um modelo com significativa informação fornecida a interessados no projeto
- Estrutura dinâmica
- Iteração



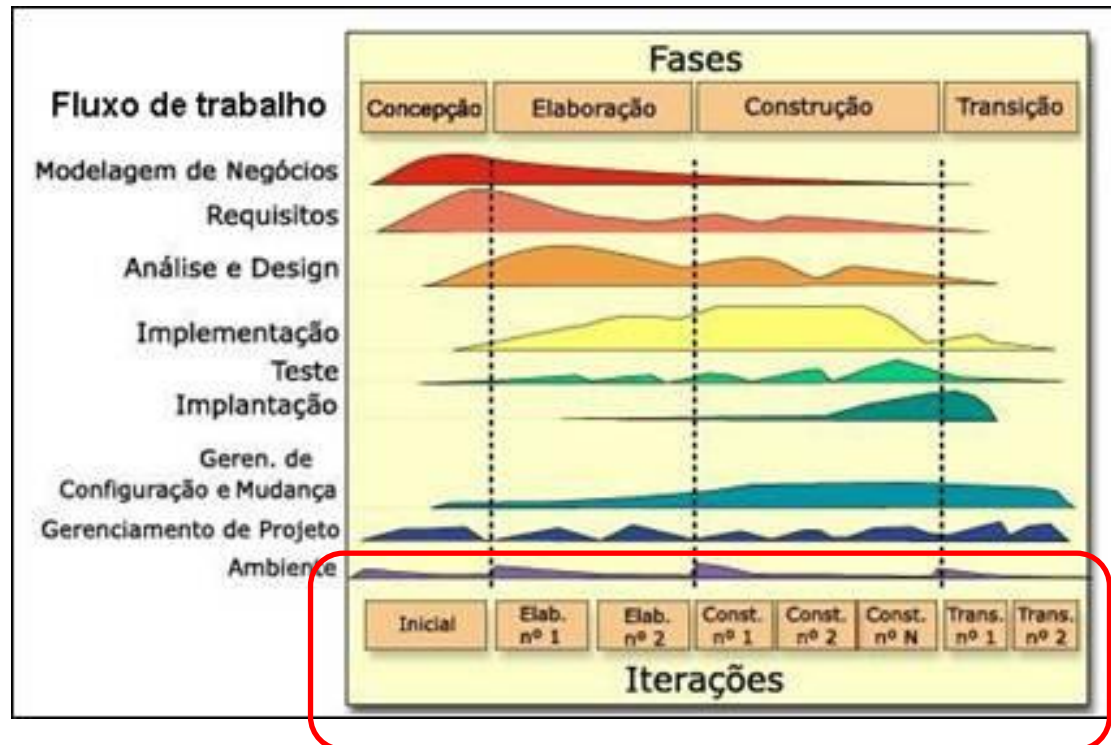
Características do Processo Unificado

- Estrutura estática
- Estrutura dinâmica
 - Estrutura do ciclo de vida em fases: concepção, elaboração, construção, transição
 - Cada fase com uma ou mais iterações
 - Um fase é o tempo decorrido entre dois marcos principais
- Iteração



Características do Processo Unificado

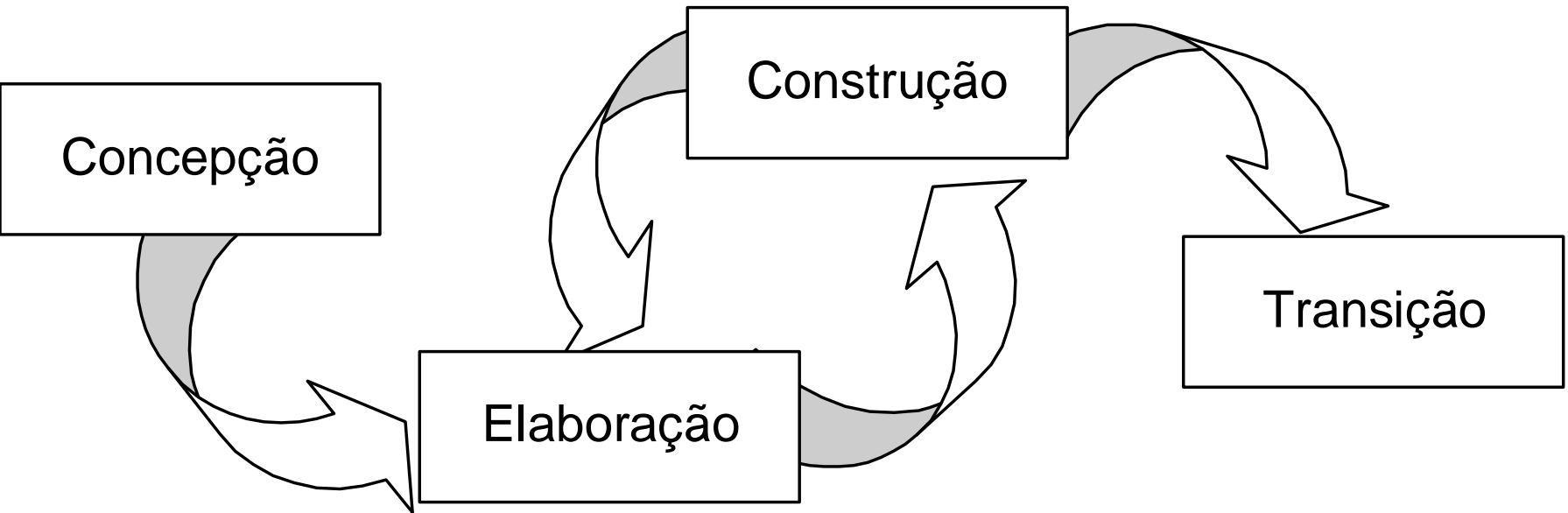
- Estrutura estática
- Estrutura dinâmica
- Iteração
 - É um mini-projeto da fase, que resulta um incremento, e atravessa os 5 workflows
 - Um incremento é uma versão com funcionalidades adicionadas ou melhoradas em comparação à versão anterior



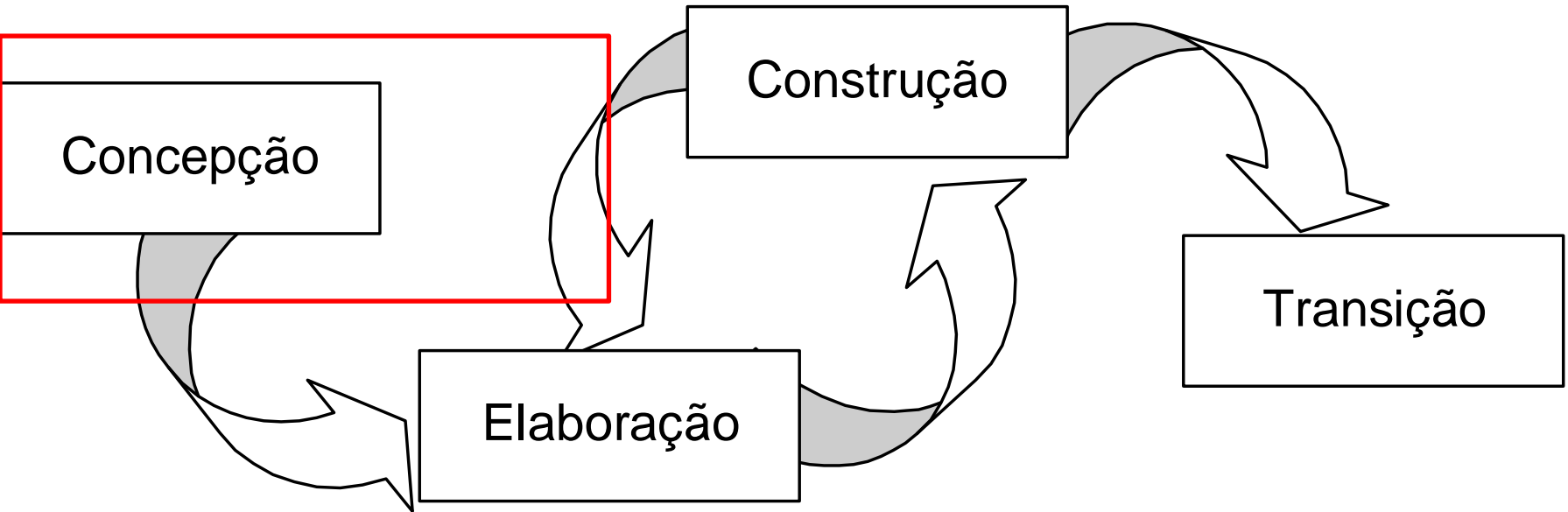
Processo Unificado

- É um processo iterativo
 - Miniprojetos curtos de duração fixa
 - Produto de cada miniprojeto é um sistema parcial, executável, testável e integrável
- Incremental
 - O sistema cresce incrementalmente
- Evolutivo
 - A cada incremento o sistema evolui
- É flexível
 - Pode ser aplicado em uma abordagem ágil e leve incluindo práticas de outros métodos ágeis como XP e Scrum
- Permite realimentação e readaptação

Ciclo de Vida: Fases



Ciclo de Vida: Fases



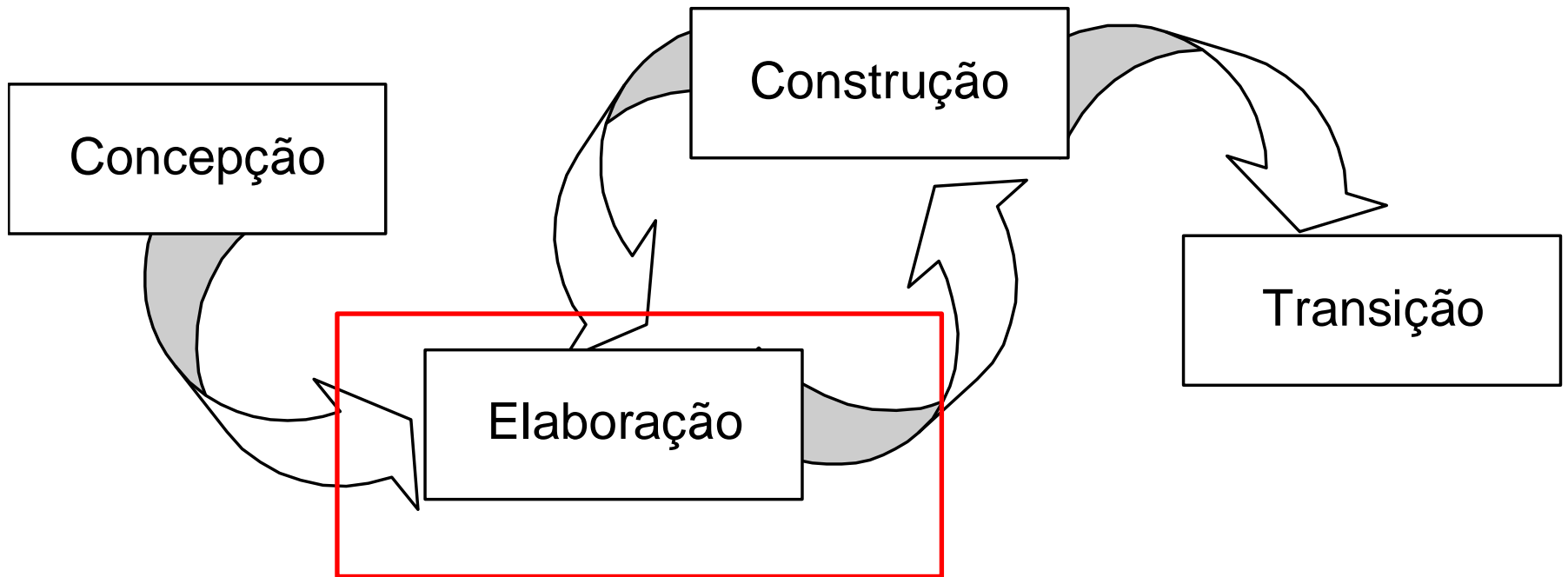
A Concepção

- Concebe o escopo do produto, a visão e o caso de negócio
- Buscar as primeiras informações sobre o sistema a ser desenvolvido
- Descobrir se vale a pena fazer a análise, mas sem fazer a análise propriamente dita
- Não obtém o plano detalhado do projeto, apenas uma visão superficial
- Os requisitos são levantados de forma superficial
- Apenas 10% destes requisitos serão detalhados
 - Os de maior risco
 - Os de maior impacto no negócio
 - Os mais significativos arquiteturalmente

Na Concepção

- Quanto tempo?
 - Estima-se 1 semana para esta etapa
 - Desta 1 semana, 2 dias para seminário de requisitos
- Como?
 - Inicia pela identificação dos interessados e objetivos
 - Escrever os objetivos como casos de uso na forma essencial e resumida
 - Escolher 10% a 20% dos casos de uso mais relevantes
 - Reescrever estes casos de uso na forma completa

Ciclo de Vida



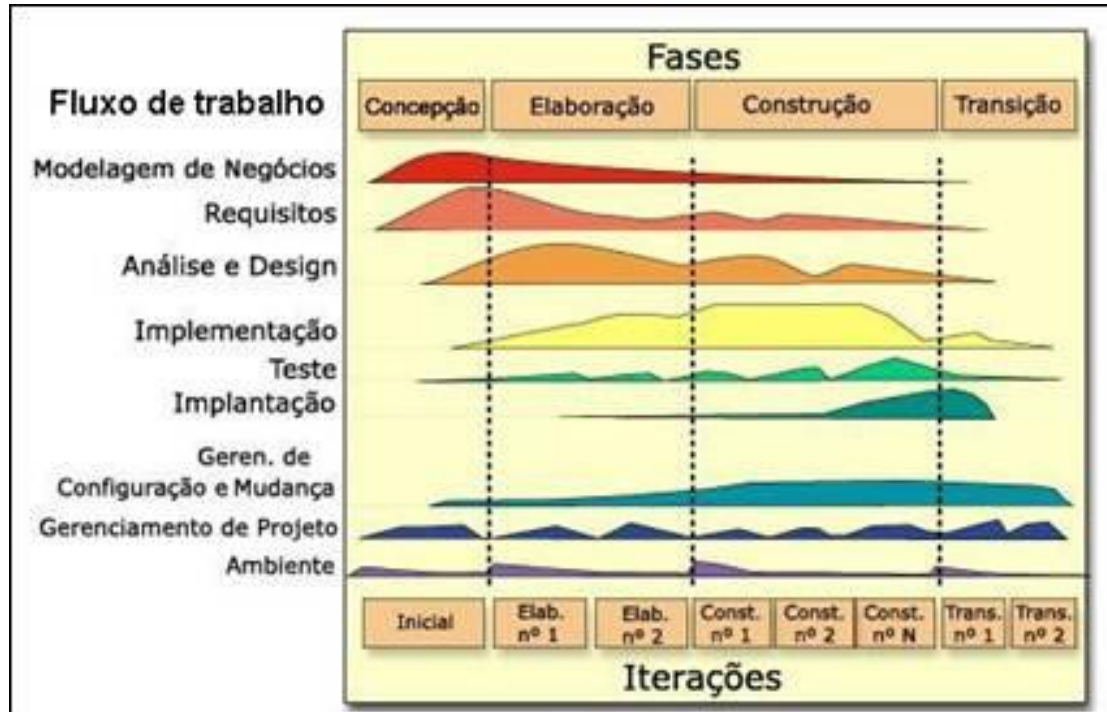
Na elaboração

- Arquitetura central e de alto risco é programada e testada
- A maioria dos requisitos é descoberta
- Os principais riscos são mitigados
- Consiste em duas ou mais iterações
- Iterações de 2 a 6 semanas

Fluxos de Trabalho do PU

- Fluxos de Processo
 - Modelagem de negócios
 - Requisitos
 - Análise e design
 - Implementação
 - Teste
 - Implantação
- Fluxos de Suporte
 - Gerenciamento de Configuração e Mudança
 - Gerenciamento de Projeto
 - Ambiente

Workflow e Fases do Processo Unificado



Exemplos de iterações

	Concepção	Elaboração	Construção	Transição
Projetos simples	1 iteração -protótipo de tela	1 iteração -arquitetura	1 iteração - versão beta	1 iteração -implantação
Projetos de média complexidade	1 iteração - entender negócio e protótipo de prova	2 iterações - capturar requisitos e propor arquitetura - finalizar arquitetura	2 iterações -funcionalidades centrais - concluir versão beta	1 iteração - fazer a implantação
Projetos complexos	2 iterações -entender negócio - um ou mais protótipos	3 iterações - capturar requisitos - esboçar arquitetura - finalizar arquitetura	3 iterações -funcionalidades principais - implementar mudanças -concluir beta	2 iterações - desenvolver versão final - fazer implantação