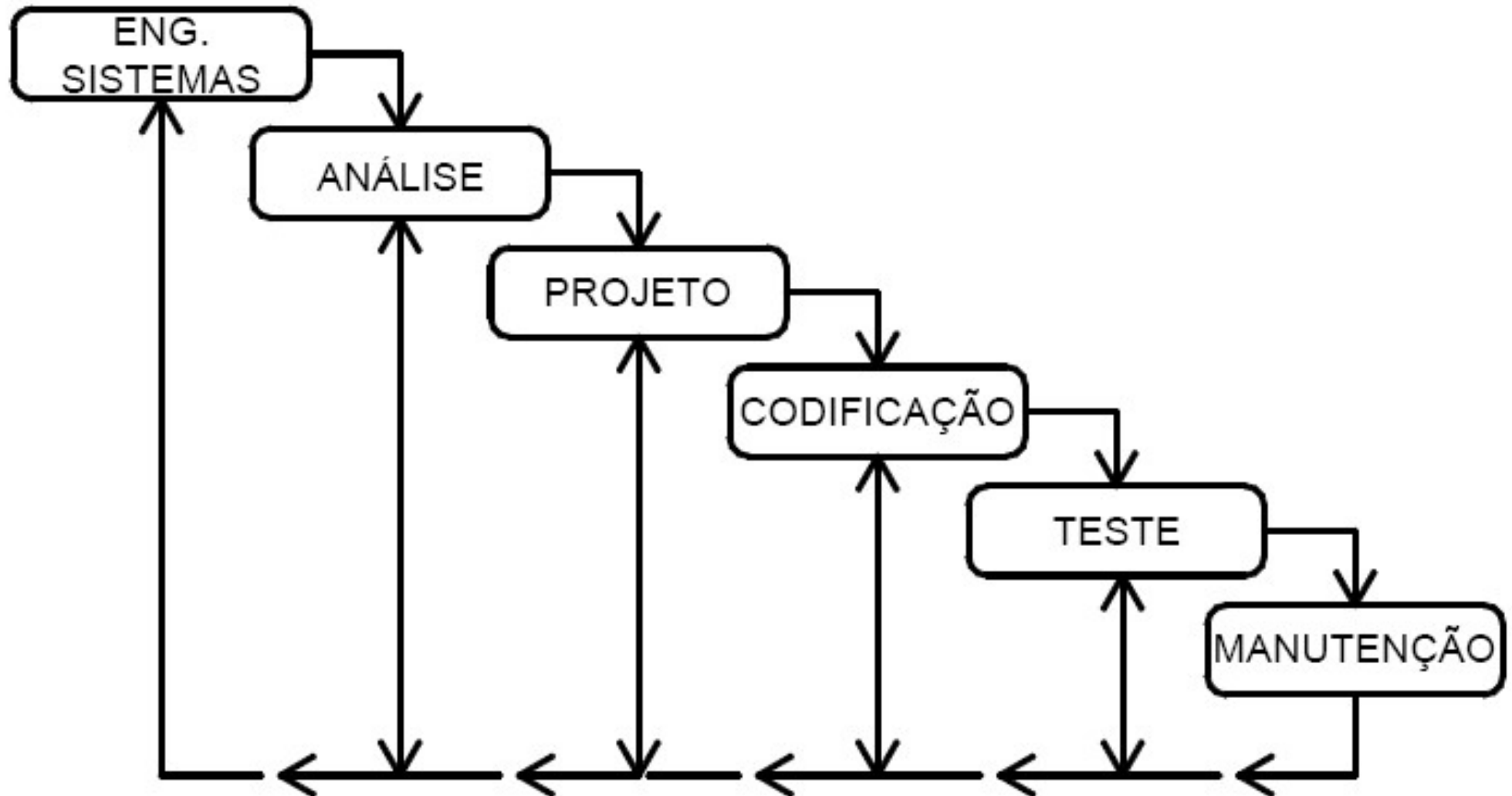


Engenharia de Software

Processo de Desenvolvimento

- Também chamado de ciclo de vida do software
- Reflete os passos necessários para se construir um produto de software
- Existem vários modelos de ciclo de vida
 - Cascata (1956)
 - Iterativo e Incremental (1968)
 - Espiral (1986)
 - RUP (1996)
 - Extreme Programming (XP) (1999)
 - Agil (2001)

Ciclo de Vida - Modelo Cascata



Engenharia de Software

□ ENGENHARIA DE SISTEMAS

- Levantamento dos requisitos
- Inserir o sistema em um contexto maior –Hardware; Pessoas; Outros sistemas
- Visão geral e ampla do sistema
- Riscos; Custos; Prazos; Planejamento

Engenharia de Software

□ ANÁLISE

- “O que” o software conterà?
- Continua o processo de coleta de requisitos, porém concentra-se no âmbito do software
- Modelos – Dados; Funções e comportamentos
- Particionamento do problema
- Domínio problema
- Documentação e Revisão dos requisitos
 - ANÁLISE ESTRUTURADA – DFD
 - ANÁLISE ORIENTADA A OBJETOS – Diagramas: Caso Uso, Atividades, Sequencia, Classes (Model)

Engenharia de Software

□ PROJETO

- “Como” o software irá executar os requisitos
- Estrutura de dados; Arquitetura do Software;
- Detalhes de execução; caracterização da interface
- Domínio solução
- Produzir um modelo que permita a sua construção posterior
 - PROJETO ESTRUTURADO – Módulos
 - PROJETO ORIENTADO A OBJETOS – Atributos; Especificação dos Métodos; Mensagens
 - Diagramas de Sequência; Diagrama de Classes

Engenharia de Software

□ CODIFICAÇÃO

- “Traduzir” o projeto para uma linguagem de computador
- Projeto detalhado pode levar a uma codificação mecânica (Ferramenta CASE)

□ TESTES

- Verificação se o código atende aos requisitos
- Aspectos lógicos e internos do software – Teste de todas as instruções
- Aspectos funcionais externos – entrada produz o resultado esperado

Modelo Cascata – Críticas

- O modelo cascata é fortemente baseado em suposições oriundas dos processos de engenharia convencionais
- Algumas dessas suposições não foram confirmadas na prática
 - Todos os requisitos podem ser precisamente identificados antes do desenvolvimento
 - Os requisitos são estáveis
 - O projeto pode ser feito totalmente antes da implementação

Modelo Cascata – Críticas

❑ Instabilidade dos Requisitos

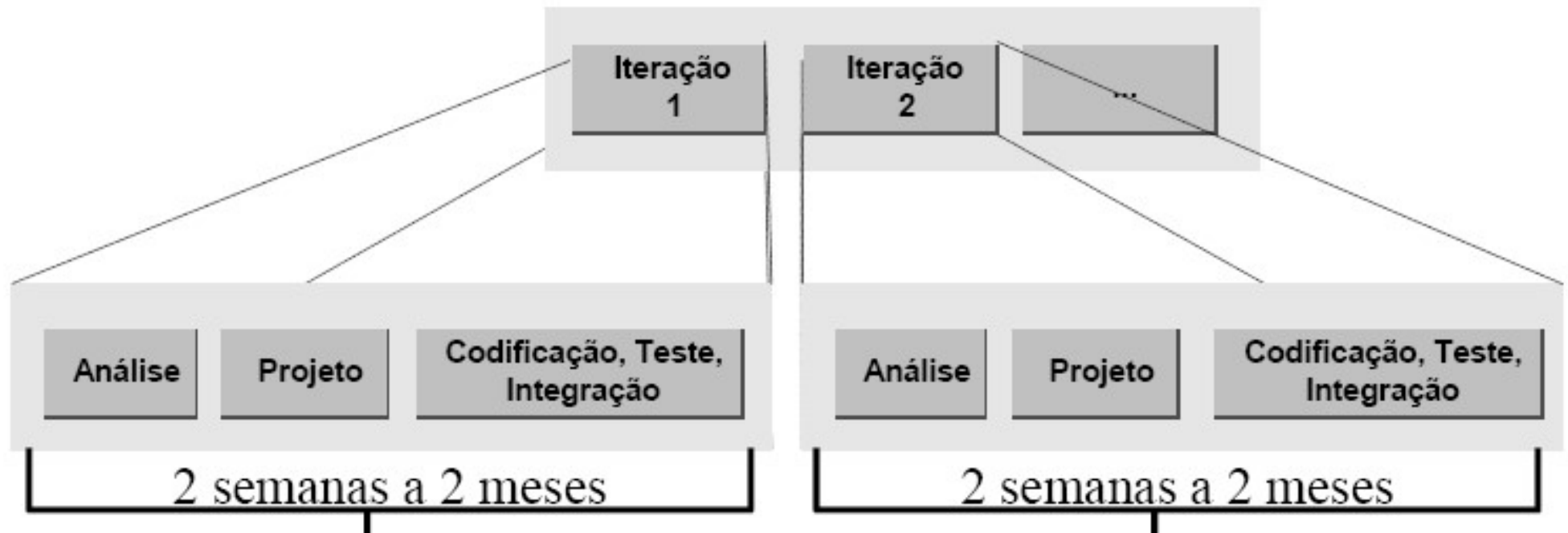
- O mercado está em mudança constante.
- As tecnologias inevitavelmente mudam e evoluem
- A vontade e objetivos dos usuários mudam, muitas vezes, de forma imprevisível.

❑ Projeto Completo antes da Implementação?

- Pergunte a qualquer programador!
- Uma especificação completa tem que ser tão detalhada quanto o próprio código
- Desenvolver software é uma atividade intrinsecamente “difícil”

Ciclo de Vida - Modelo Iterativo

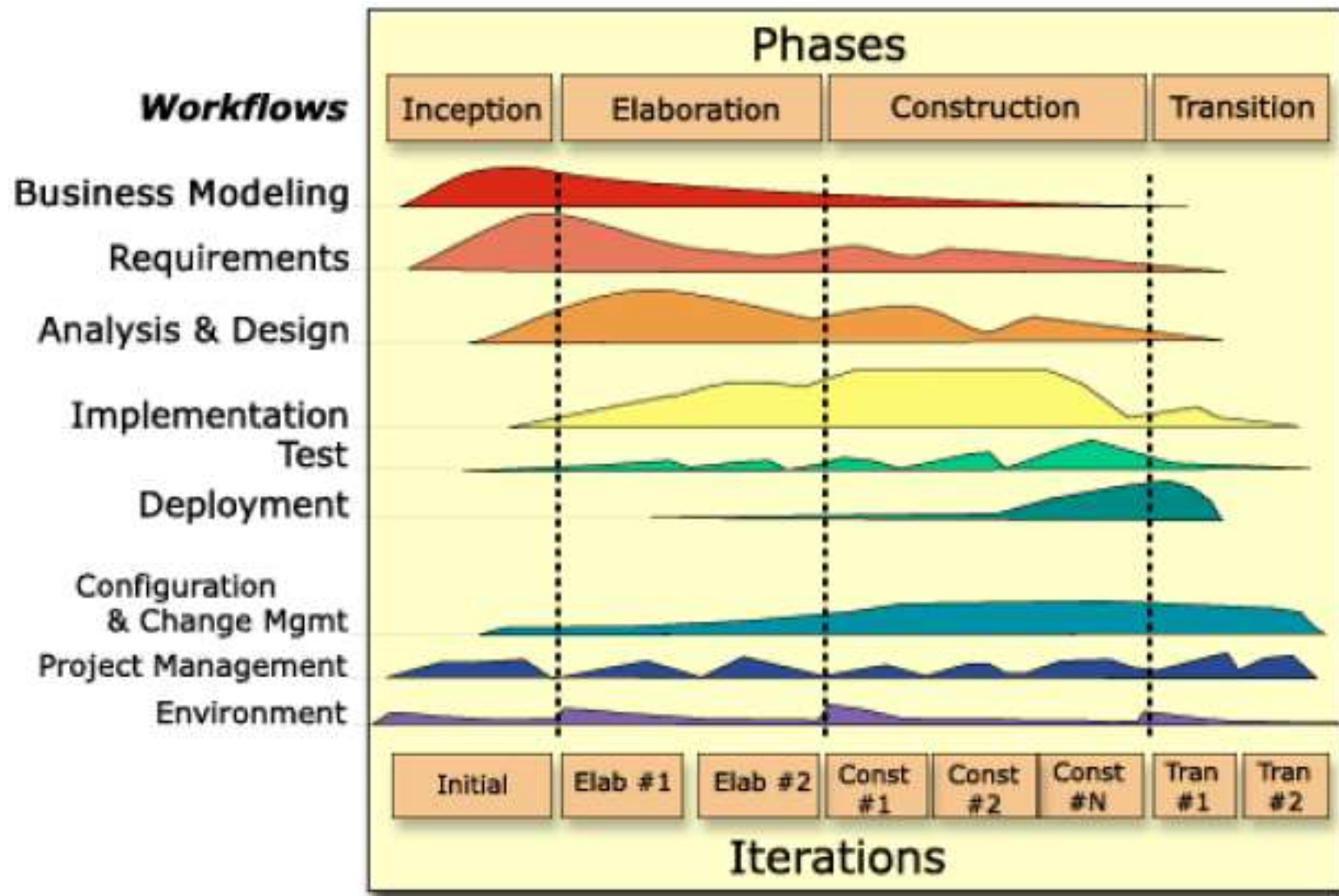
- Passos curtos, feedback e refinamento
- Iterativo, incremental, com intervalos de tempo (ciclos) pré-estabelecidos



Modelo Iterativo

- Baseia-se no fato de que não se deve ter o software inteiro funcionando por inteiro no primeiro release. Isto é um grande risco!
- Um processo de desenvolvimento deve ser:
 - Iterativo - Ter várias iterações no tempo. A iteração dura entre 2 semanas e 2 meses
 - Incremental - Gerar novas versões incrementadas a cada release.
- A cada iteração aumenta a compreensão do problema e são introduzidos aperfeiçoamentos sucessivos.

Modelo Iterativo - RUP

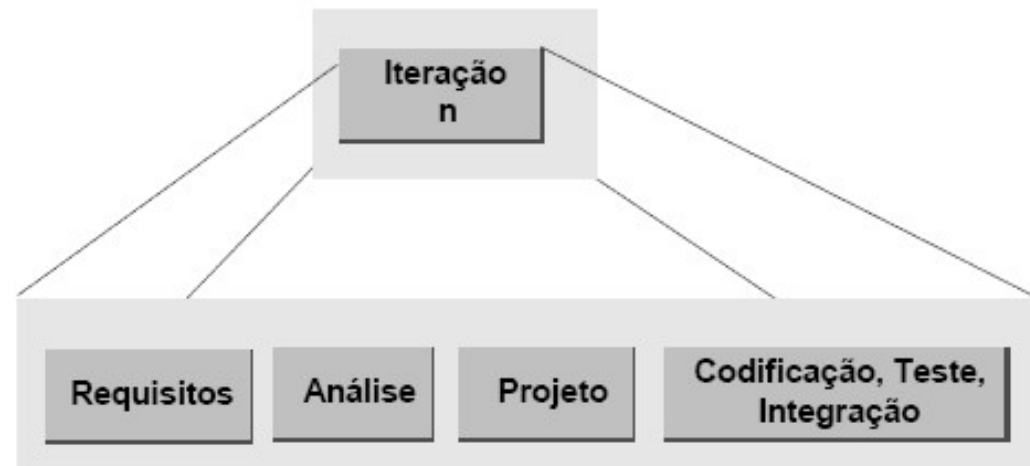


Modelo Iterativo - RUP

- ❑ No RUP (Rational Unified Process) a ênfase é dada na criação de MODELOS (UML) ao invés de documentos.
- ❑ As atividades de desenvolvimento são orientadas por caso de uso.
- ❑ O RUP encoraja o controle de qualidade e o gerenciamento de riscos, contínuos e objetivos
- ❑ O desenvolvimento é dividido em **FASES e ITERAÇÕES**.
 - FASE – Período de tempo entre marcos do processo, onde um conjunto bem definido de objetivos é alcançado.
 - ITERAÇÃO – Em cada fase, ocorrem várias iterações

Modelo Iterativo - RUP

- Em casa FASE acontece várias **ITERAÇÕES**
 - Uma iteração equivale a um ciclo completo de desenvolvimento
 - Cada iteração resulta em um projeto executável
 - Ao final de cada iteração é possível avaliar se as metas foram alcançadas e caso seja necessário é possível reestruturar o projeto



Modelo Iterativo – RUP - Fases

□ **Concepção**

- Estabelece os casos de negócio para o projeto e delimita o escopo do projeto.
- Os Casos de negócio incluem: Critérios de Sucesso; Avaliação de Riscos; Recursos Necessários. Durante a concepção é comum a criação de um protótipo executável, utilizado como testes para concepção.
- Ao final desta fase deve ser feita a decisão de continuar ou não o desenvolvimento

□ **Elaboração**

- Suas metas incluem: Análise do problema; Estabelecimento de uma arquitetura sólida; Eliminação de elementos de mais alto risco.
- É necessário a maioria dos requisitos do sistema.
- A implementação deve mostrar escolha da arquitetura.

Modelo Iterativo – RUP - Fases

□ Construção

- Desenvolvimento de maneira iterativa e incremental um produto completo.
- Descrevendo os requisitos restantes e critérios de aceitação. Nesta fase o sistema ganha corpo, conclui-se a implementação e a realização de testes do software.
- Ao final desta fase deve ser decidido se o software, ambientes e usuários estão prontos para se tornarem operacionais.

□ Transição

- Fornece o sistema a seus usuários finais.
- Esta fase é tipicamente iniciada pelo fornecimento de uma versão beta.
- Nesta fase são feitos desenvolvimentos adicionais a fim de ajustar o software.

RUP – Fluxos de Trabalho

- Modelagem do Negócio
 - Descreve a estrutura e a dinâmica da empresa
- Requisitos
 - Identificação dos requisitos a partir de casos de uso (Use Cases)
- Análise e Projeto
 - Descreve as várias visões da arquitetura através de modelos UML
- Implementação
 - Desenvolvimento do software; Teste Unitários e Integração
- Teste
 - Casos de teste; procedimentos e medidas para acompanhamento de erros
- Implantação
 - Configuração do sistema a ser entregue

RUP – Fluxos de Trabalho

- Gerenciamento da Configuração
 - Controle de modificações
- Gerenciamento do Projeto
 - Descreve as estratégias para o trabalho com o processo iterativo
- Ambiente
 - Infra-estrutura necessária para o desenvolvimento do sistema