# Introdução Engenharia de Requisitos

## Problemas do Processo

#### Problemas do Processo de Desenvolvimento de Sistemas

- Estimativas de prazo, esforço e custo imprecisas.
  - Sistemas com atraso e custando mais caro do que o previsto.
- Qualidade do software às vezes é menor que a adequada
  - causando insatisfação do usuário final, pois o sistema informatizado entregue:
    - não faz o que deveria fazer;
    - é pouco confiável;
    - é lento;
    - é difícil de usar;



#### Problemas do Processo de Desenvolvimento de Sistemas

- Alterações de metas e estratégias organizacionais
  - Os negócios requerem ciclos de desenvolvimento mais curtos
  - Requisitos iniciais fracamente definidos
- Falha no gerenciamento de riscos
  - Não há garantias que o sistema irá funcionar
  - O resultado é um grande risco
- Complexidade do software
  - Demanda crescente por novos softwares
  - Complexidade do domínio
  - Mudanças de equipe



#### **Algumas Causas dos Problemas**

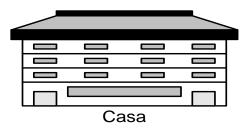
- Falta de processos e métodos.
- Métricas sobre produtividade inexistentes ou insuficientes,
  - comprometendo as estimativas de prazo, esforço e custo;
  - não permitindo a avaliação de novas ferramentas, técnicas e padrões.
- Documentação insuficiente.

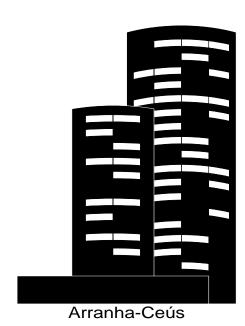


### A Importância da Análise e do Projeto

Uma analogia...







Aumento da complexidade



#### O que é um Processo?

■ O que é um Processo?

"Um processo é uma seqüência de passos realizados com um propósito. Ou seja,processo é o que você faz. O processo integra pessoas, ferramentas e procedimentos. Processo é o que pessoas fazem, utilizando procedimentos, métodos, ferramentas, e equipamentos, para transformar matéria prima (inputs) em um produto (outputs) de valor para os seus clientes."

(SEI - CARNEGIE MELLON UNIVERSITY. The Capability Maturity Model: guidelines for improving the software process. Addison Wesley, 1995. p. 8.)



## O que é um Processo de Desenvolvimento de Software?

"Um processo de desenvolvimento de software pode ser definido como um conjunto de atividades, métodos, práticas e transformações que pessoas utilizam para desenvolver ou dar manutenção em softwares ou em seus produtos associados (projetos, manuais, código, etc.)"

(SEI - CARNEGIE MELLON UNIVERSITY. The Capability Maturity Model: guidelines for improving the software process. Addison Wesley, 1995. p. 8.)

### 10

#### O que se pretende com a modelagem?

- Capturar os processos de negócio
- Facilitar a comunicação entre as partes envolvidas (especialistas e analistas)
- Facilitar a gerência da complexidade do domínio permitindo exibir várias visões dos elementos de modelo
- Definir a arquitetura lógica independente das possíveis implementações
- Permitir o reutilização de esforços

## м

#### Importância da modelagem

- Ajuda a visualizar um sistema como ele é, ou como se deseja que ele seja
- Permite especificar o comportamento e a estrutura de um sistema, ou seja, a sua arquitetura
- Oferece uma representação que serve como guia construção do sistema
- Documenta as decisões tomadas
- Diminui o risco de implementações erradas



#### Princípios da modelagem

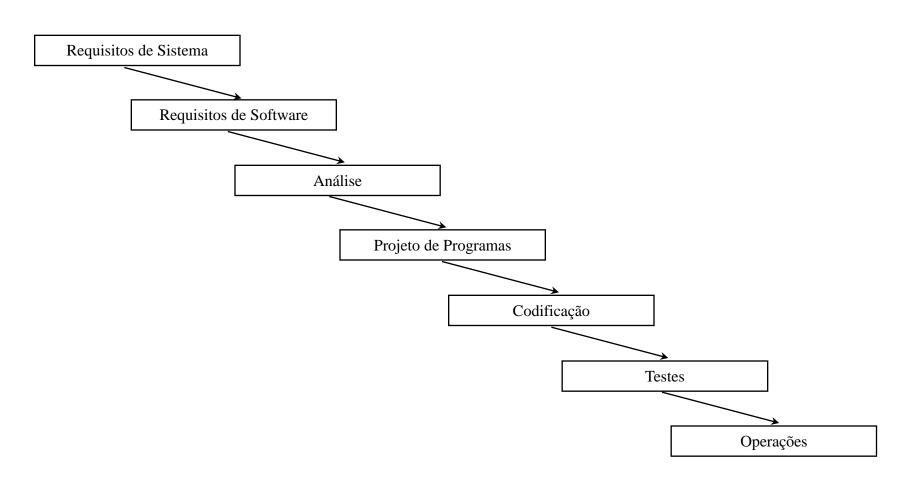
- Quanto mais complexo o sistema, maior a necessidade de projeto (Ex: casa de cachorro, sítio, prédio de 10 andares)
- Cada modelo poderá ser expresso em diferentes níveis de detalhamento
- Conforme Booch 'Um modelo é uma simplificação da realidade.'
- Nenhum modelo único é suficiente
- Abstração como técnica para sistemas complexos, estudando partes e esquecendo temporariamente a complexidade do problema.

#### .

#### Tipos de Processos de Software

- Cascata
- Prescritivo (UP)
- Ágil

## Modelo de Desenvolvimento de Sistemas em Cascata (Projeto Clássico)



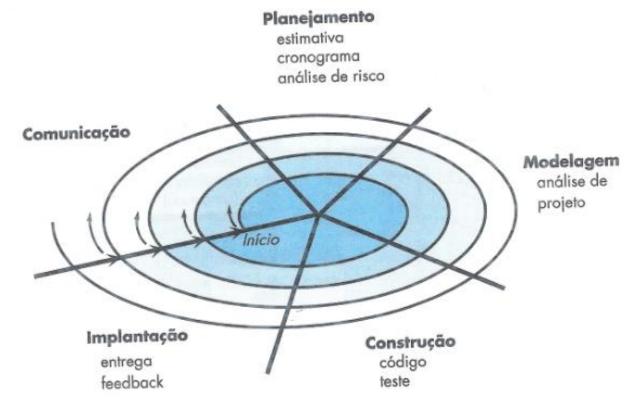


#### **Modelo Cascata**

- Baseado na linha de montagem
- Difícil de lidar com mudanças de requisitos
- Torna difícil a entrega parcial de produtos
- Progressão sequencial entre uma fase e outra
- O modelo não suporta revisões das fases
- Usuário visualiza o sistema somente no final: baixa interação
- Implementação bottom-up: testes dos módulos, dos subsistemas e dos sistemas nessa ordem
- Adia a identificação de riscos, tornando extremamente caro desfazer algum erro cometido nas fases iniciais

### **Modelo Espiral**

- Desenvolvimento incremental
- Regiões de tarefas que variam de acordo com o processo de software da empresa:



## 7

# Processo Racional Unificado (Rational Unified Process - Rup)

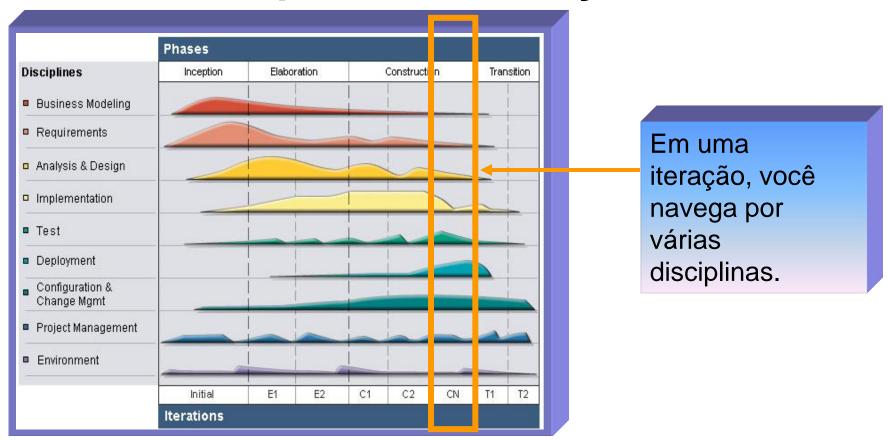
- Um processo visa atingir uma meta
- Na engenharia de software um processo visa entregar um produto eficiente e previsível que atendam as necessidades de um negócio
- RUP consiste de uma abordagem do ciclo do processo de desenvolvimento de sistema adequada à UML.
- Objetiva assegurar a produção de softwares de qualidades que atendam aos usuários e possuam orçamento e cronogramas bem definidos.
- É um processo Iterativo e incremental



## O que é o desenvolvimento iterativo e incremental?

- É o processo de construção de sistemas de software feito em pequenos passos.
- Permite reavaliações e readequações.
- Requer uma compreensão crescente do domínio por meio de aperfeiçoamento sucessivos.
- Identificação de riscos no início do projeto
- Benefícios:
  - Redução do risco devido a uma percepção prematura do domínio.
  - Aumento da flexibilidade de alteração/manutenção
  - Melhoria na qualidade do produto

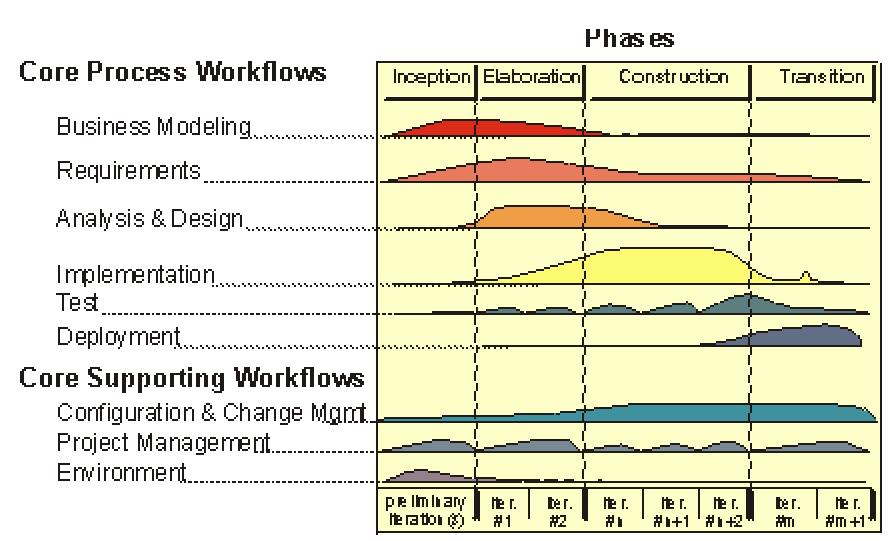
#### O que é uma Iteração ?



Iteração: consiste em uma sequência de atividades distintas com um plano e um critério de avaliação, resultando em uma versão.

### M

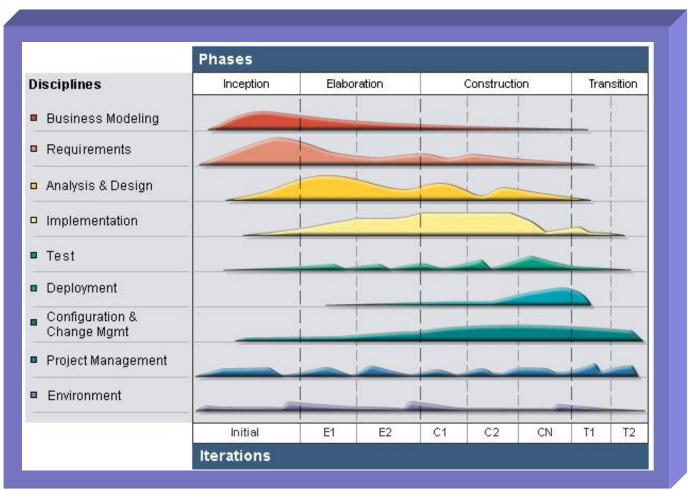
#### Dimensões do RUP



Iterations

#### Organização ao longo do Tempo

Tempo

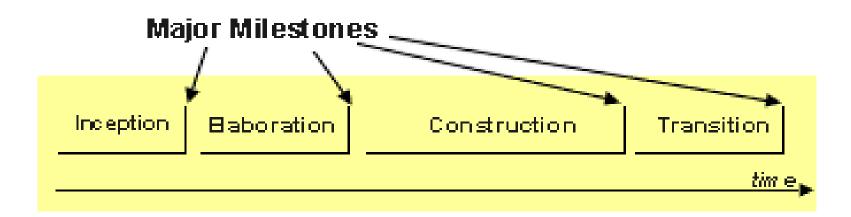


Organização por fases minimiza os riscos da alocação de recursos.



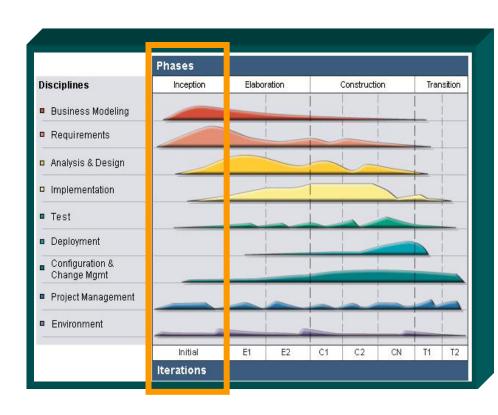
#### Dimensões do RUP

- O RUP pode ser descrito em duas dimensões:
- TEMPO mostra os aspectos dinâmicos do processo, os ciclos, fases, iterações e marcas (milestones)
  - Fase período de tempo entre dois importantes marcos de um processo
  - □ Uma fase é dividida em iterações
  - □ Fases do RUP : Concepção, Elaboração, Construção e Transição





- Estabelecer escopo de projeto e condições de limite
- Determinar casos de uso preliminares e cenários que podem restringir o projeto
- Elencar uma arquitetura (plataforma HW/SW) candidata
- □ Estimar o custo e prazo globais
- Identificar riscos potenciais
- Preparar o ambiente de suporte ao projeto



#### Fase de Concepção

#### Objetivo :

 Estabelecer qual o negócio para o novo sistema ou para uma atualização de um sistema já existente e delimitar o escopo do projeto

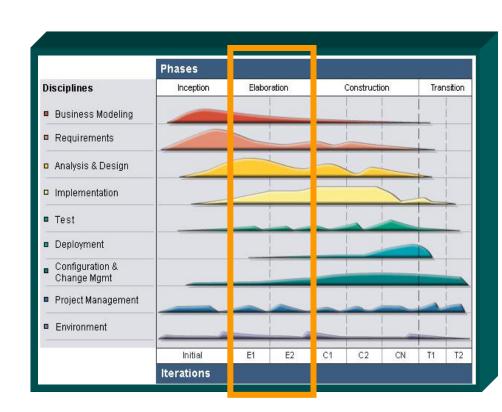
#### Alguns produtos:

- avaliação inicial de riscos
- estimativa de recursos necessários
- o início da construção do caso de uso
- um plano do projeto mostrando fases e iterações
- levantamento dos critérios de sucessos para o negócio



#### Fase de Elaboração: Objetivos

- Definir e validar a arquitetura
- Atacar riscos arquiteturais significativos
- Especificar um plano detalhado para a fase seguinte de Construção
- Demonstrar que a arquitetura escolhido suporta o sistema com um custo razoável
- Organizar o ambiente de suporte ao projeto





#### Fase de Elaboração

- Obter um melhor entendimento do problema
- Definir o que construir, como construir e a tecnologia a utilizar.
- Desenvolve-se um plano de como será realizado o projeto.
- Alguns produtos:
  - □ caso de uso quase completo
  - □ descrição da arquitetura do software
  - □ reavaliação da lista de riscos,
  - plano completo do desenvolvimento dos sistema e descrição dos diversos releases do sistema.

