

# RUP/PSDS



## Introdução e Comparação

- RUP
  - Introdução
  - Melhores Práticas
  - Estrutura
    - Tempo
    - Conteúdo
- Contraponto PSDS



- Objetivos
  - Promover planejamento, medição e controle dos projetos
  - Reduzir riscos
  - Fazer do desenvolvimento algo previsível, medível e repetível
- Natureza Empírica

- Implementadas no RUP
- Suportam a realização dos objetivos do processo
- Conjunto de abordagens comprovadas comercialmente para resolver grandes problemas da Eng<sup>o</sup> de Software



- Necessidades não são satisfeitas
- Requisitos são mal definidos
- Modulos não se integram
- Dificuldade de manutenção
- As falhas são descobertas tarde demais...



# Melhores Práticas

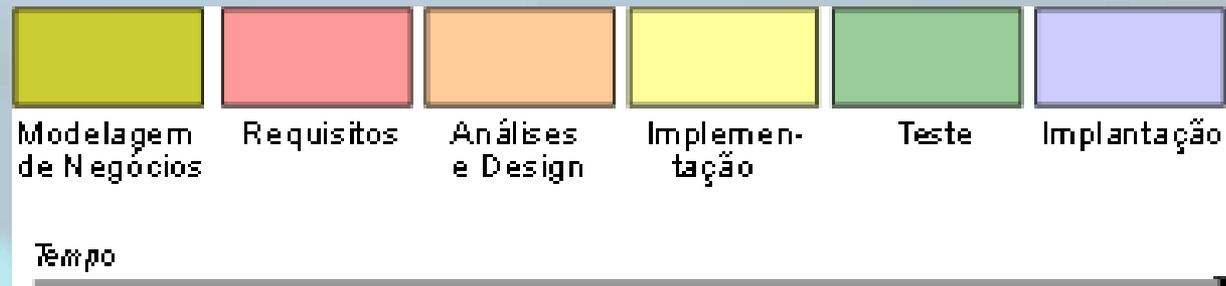
- Desenvolva Iterativamente... adiantando integração das unidades e ataque aos riscos
- Gerencie Requisitos... garantindo o envolvimento do usuário
- Use uma Arquitetura de Componentes... validando decisões arquiteturais mais cedo
- Modele visualmente... administrando complexidade
- Verifique Qualidade Continuamente... diminuindo erros e custo
- Gerencie Mudanças... evoluindo baselines incrementalmente



- Tempo
  - Ciclo de vida Iterativo e Incremental
  - Fases, Marcos e Iterações
- Conteúdo
  - Disciplinas: Papéis, Artefatos e Atividades

# Modelo Cascata

- Antigamente as disciplinas eram executadas em sequencia numa única vez



- Integração tardia... grandes surpresas em requisitos, projeto, implementação!

# Modelo Iterativo

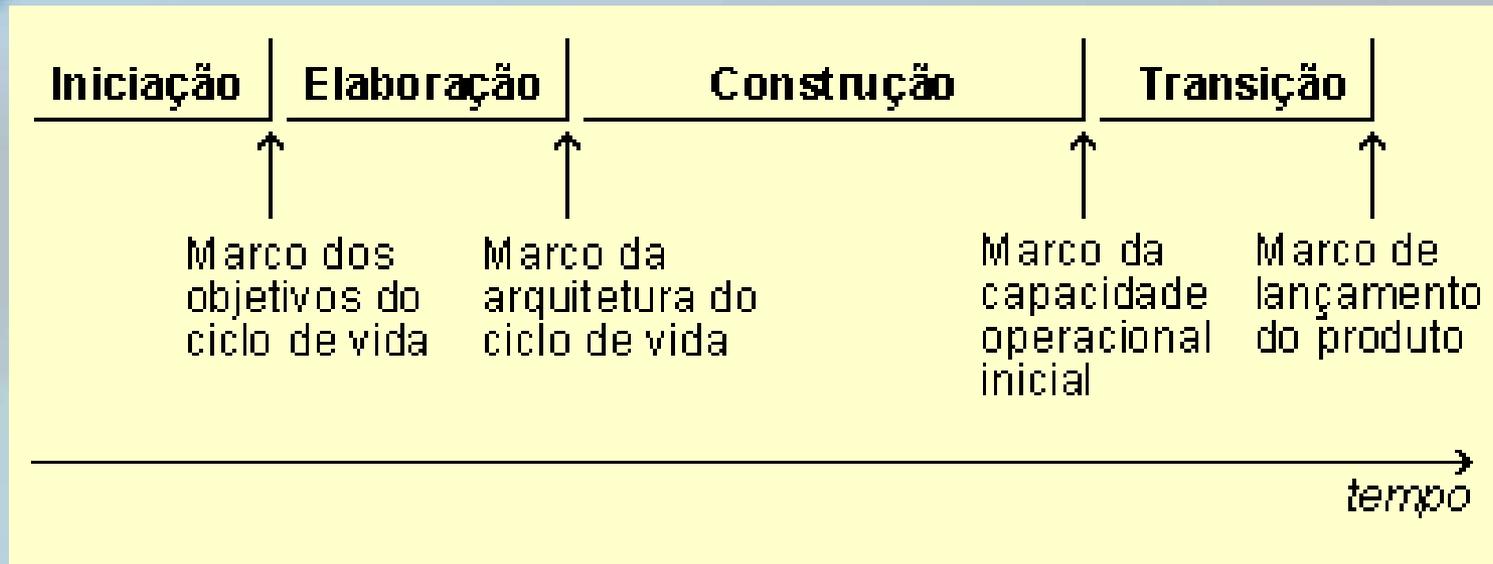
- Cada iteração libera um produto executável estável
- Iterações incrementam o produto



- Suporta mudanças
  - Requisitos
  - Tecnologia
- Integração e testes contínuos
- Riscos são atacados logo
- Time to Market
- Aumenta qualidade
- Gestão Controlada



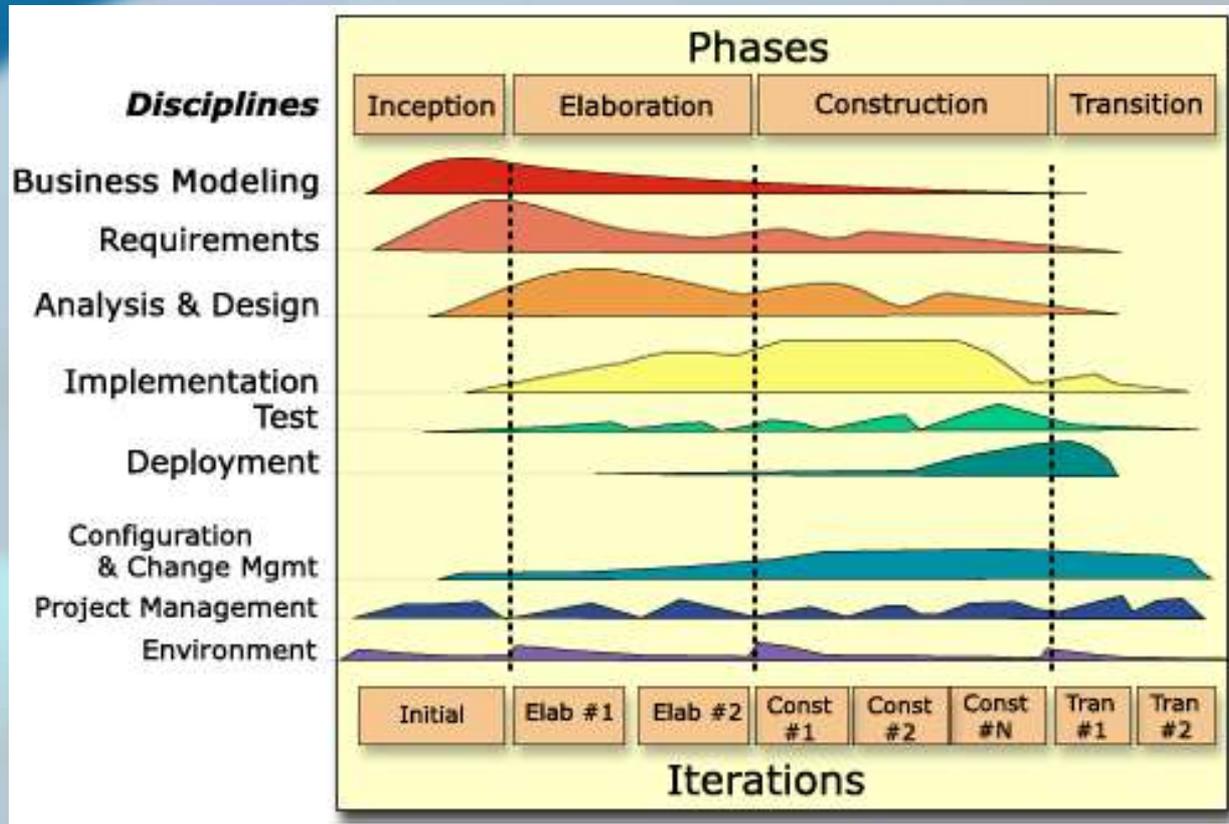
# Fases, Marcos e Iterações





- Determina escopo
- Estima custos e cronograma
- Identifica riscos potenciais
- Especifica casos de uso e cenários mais importantes
- Demonstra uma arquitetura candidata

# Concepção



# Marco: Objetivos do Ciclo de Vida

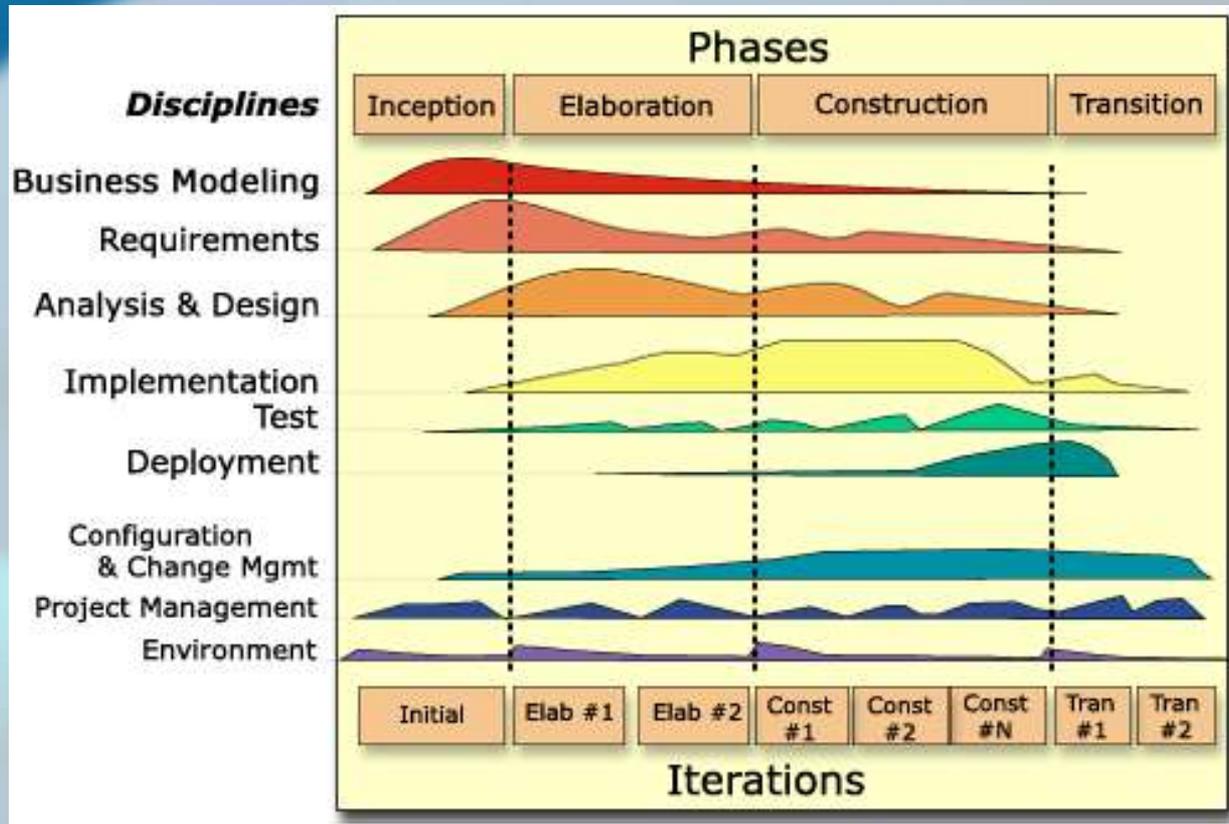
---

- Requisitos capturados e bem entendidos
- Acordado que estimativas de custos, cronograma, riscos e prioridades estão OK





- Construir uma arquitetura robusta
- Trazer um plano detalhado para a construção



# Marco: Arquitetura do Ciclo de Vida

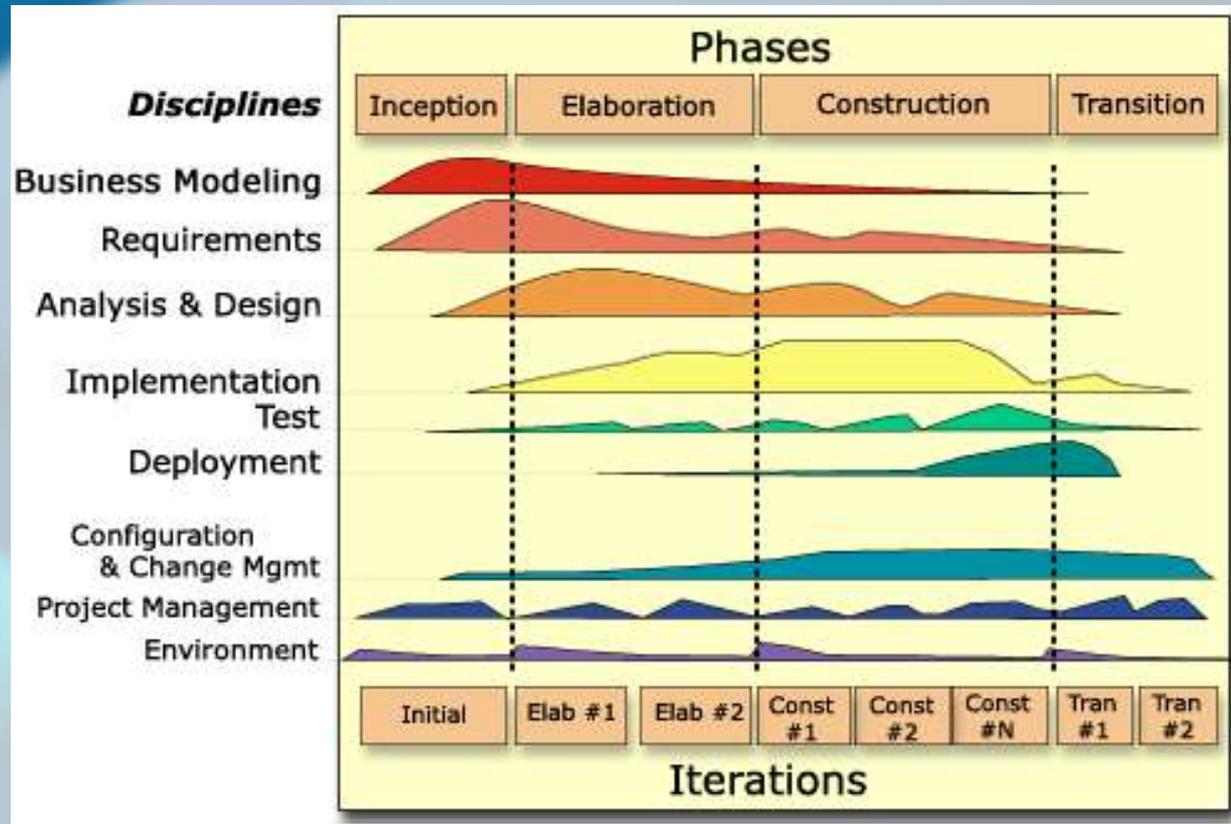
---

- Visão e Requisitos estão estaveis
- Arquitetura esta estavel
- Maiores riscos já foram tratados





- Completar a construção do software para colocá-lo em produção
- Fase industrial do processo



# Marco: Capacidade Operacional Inicial

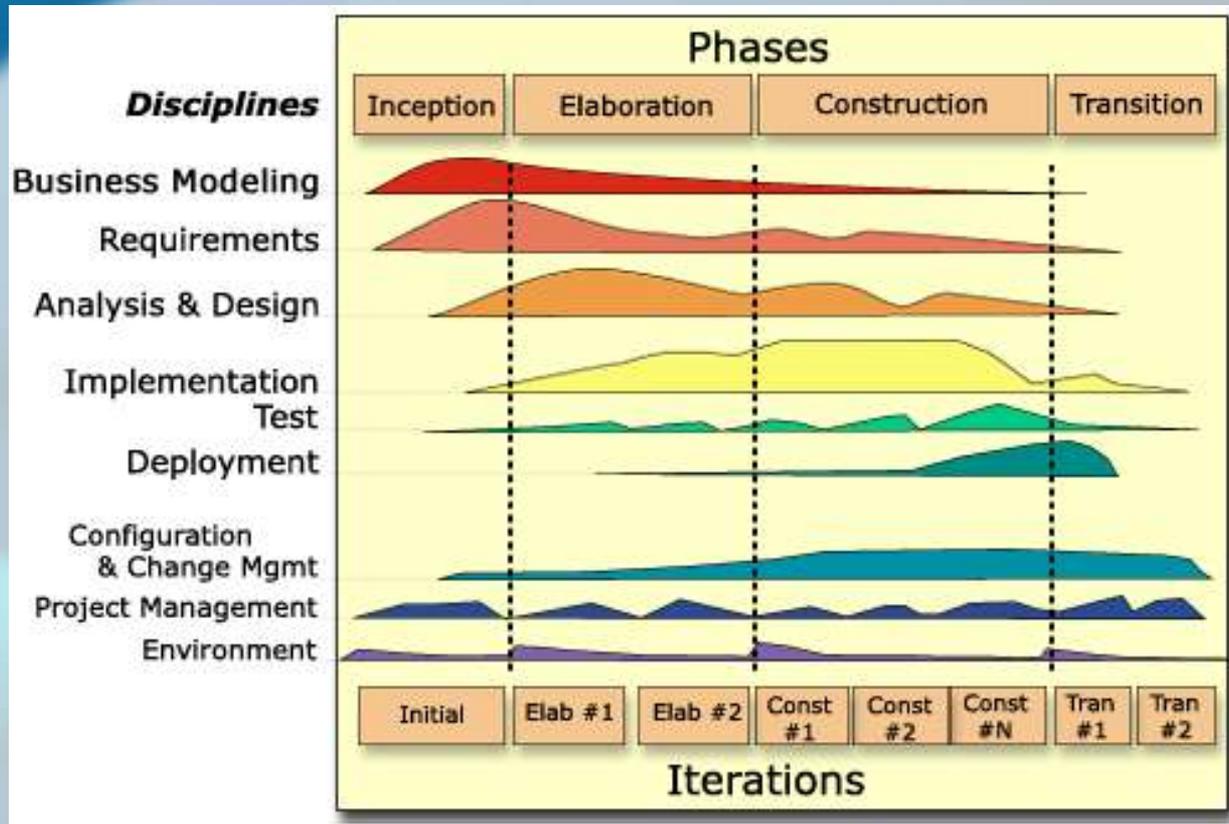
---



- O produto pode ser colocado em produção
- Todos estão prontos para colocá-lo em produção



- Treinamentos
- Garantir que as especificações foram satisfeitas
- Chegar ao fim do processo de desenvolvimento



# Marco: Lançamento do Produto

---

- O usuário está satisfeito
- O que foi gasto perante o planejado está aceitável





- Um passo ao longo de todas as disciplinas – mini-projeto cascata
- Podem haver uma ou mais iterações para cada fase
- Servem como ponto de avaliação técnica

# Iteração: Tamanho e Duração

- Determinados
  - Tamanho da organização
  - Tamanho do projeto
  - Conhecimento do negócio
  - Conhecimento da tecnologia
- No total 6 com + ou – 3
  - Concepção: 0 .. 1
  - Elaboração: 1 .. 3
  - Construção: 1 .. 3
  - Transição: 1 .. 2

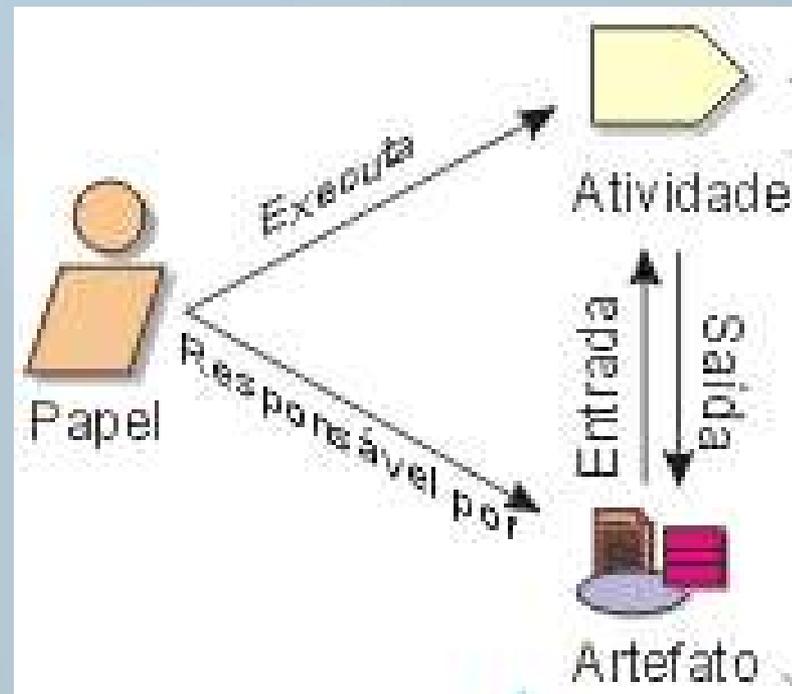




- Possuem um fluxo semi-ordenado para execução do processo
- Facilitam o entendimento do processo
- Principal objetivo é a construção de um ou mais modelos

# Principais Componentes

- Quem (Papel) faz o quê (Artefato) e como (Atividade)



- Para cada individuo ou time define
  - Comportamento
  - Responsabilidade
- Uma pessoa pode representar vários papéis
- Ex.: Analista, Arquiteto, Testador...

- Um pedaço de trabalho de algumas horas ou dias
- Tem um papel responsável
- Repetida (se necessário) a cada iteração
- Ex.: Executar Teste, Desenvolver Plano de Iteração...

- Documento ou modelo produzido ou utilizado no processo
- Pode agregar outros artefatos
- Ex.: Código Fonte, Modelo de Requisitos, Plano de Projeto...



# Outros Componentes

---

- Fluxo de Trabalho
  - Caminho em alto nível de uma disciplina
- Guias
  - Apoio para execução das atividades
- Templates
  - Modelos de artefatos



- Modelagem de Negócios
- Requisitos
- Análise e Design
- Implementação
- Teste
- Implantação
- Gerenciamento de Configuração e Mudança
- Gerenciamento de Projeto
- Ambiente

# Gerenciamento de Projeto

---

- Proporciona uma forma de planejar, montar equipes, executar e monitorar projetos
- Arcabouço para gerenciamento de Riscos



# Modelagem de Negócios

- Entender a estrutura e dinâmica da organização
- Identificar problemas e prover recomendações
- Garantir que o usuário/cliente entende a organização
- Principais Artefatos: Business Use-case Model

- Estabelecer um acordo em os interessados do que o sistema deve fazer
- Promove uma base para estimativas de custo e tempo
- Principal Artefato: Use-Case Model



- Transformar requisitos em um desenho do que será o sistema
- Alcançar uma arquitetura robusta
- Principal Artefato: Design Model (Diagramas UML)



# Implementação

- Criar um Sistema Executável
- Implementar o Design Model
- Testar as Unidades Implementadas
- Principal Artefato: Implementation Model



- Garantir a validação do sistema com o srequisitos
- Gerenciar os defeitos do sistema
- Principais Artefatos: Test Plan, Test Data e Test Results

- Garantir a disponibilidade do produto ao usuário final
- Foco em testes (no ambiente final), documentação e distribuição



# Gerencia de Configuração e Mudanças

- Rastrear e Manter a integridade dos artefatos
- Controle de Mudanças
- Versionamento



- Suportar o desenvolvimento da organização através de processos e ferramentas
- Configuração, Implementação e Melhoria do Processo
- Gerenciando as Mudanças da Organização
- Desenvolvimento do Ambiente



- Modelagem de Negócios
- Requisitos
- Análise Orientado a Objetos
- Análise Estruturado
- Projeto Orientado a Objetos
- Projeto Estruturado
- Implementação
- Testes
- Homologação
- Implantação
- Gestão de Configuração
- Gestão de Projeto
- Gestão de Subcontratação
- Garantia da Qualidade de Software

- Suporte ao paradigma estruturado
  - Hoje as macroatividades estão sendo juntadas todos utilizarão UML
  - Suporte se dará através de guias de adaptação, todos utilizam UML
- CMM
  - SubContratação
  - Garantia da Qualidade de Software
- Não existe Ambiente



# Implantação e Execução

- Desde de 2000
- O processo se dá em níveis diferentes dentro da empresa
- Hoje em processo de evolução para CMM3
  - Ajuste das atividades de Eng<sup>o</sup>
  - Inclusão de atividades para gerenciamento do processo (mensuração, planejamento e controle)



- Processo mais bem-sucedido do mercado
- Bem suportado pelo comunidade (consultores, bibliografia...)
- Escopo amplo e bem detalhado
- Pesado
- Difícil de ser implantado em pequenas empresas



- Rational. Rational Unified Process Fundamentals – Student Manual. 2002
- Quality. Fundamentos do RUP. s.d.
- Rational. Rational Unified Process. 2002

