

Estrutura e Conteúdo do RUP

A thick, horizontal yellow brushstroke underline that spans the width of the slide, positioned below the first line of text.

Fundamentos do Rational Unified  
Process

## Objetivos: A estrutura da RUP

---

- ◆ Explicar o papel do processo no desenvolvimento de software
- ◆ Explorar a organização das fases no RUP e seu relacionamento com o desenvolvimento iterativo
- ◆ Explorar a organização das disciplinas do RUP e seu relacionamento com o desenvolvimento iterativo

# Conteúdo

---

- O papel de um processo de desenvolvimento de software
- ◆ A organização das fases no RUP
- ◆ A organização das disciplinas no RUP

# Um processo de desenvolvimento

---

- ◆ Define os passos para as entregas e define quem são os responsáveis por elas
- ◆ Ajuda a controlar o projeto
- ◆ Ajuda o gerenciamento de projeto a planejar recursos e medir progresso  
Reduz o risco
- ◆ Torna o desenvolvimento de software previsível, repetido e mensurável
- ◆ Não é simplesmente um outro processo

# Rational Unified Process

---

- ◆ Fornece diretrizes para o desenvolvimento eficiente de software de qualidade
- ◆ Reduz risco e aumenta a previsibilidade
- ◆ Captura e apresenta as melhores práticas
  - Fornece aprendizagem através da experiência de outros
  - Age como um mentor no seu desktop
  - Fornece uma extensão do material de treinamento
- ◆ Promove visão e cultura comuns

## O papel da UML no RUP

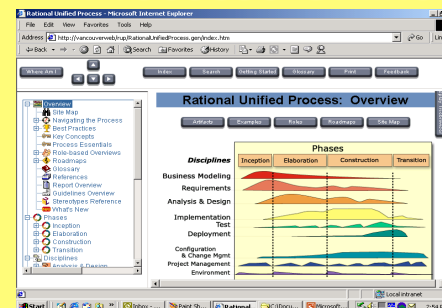
---

- ♦ Rational Unified Process foi desenvolvido paralelamente à UML.
- ♦ Muitos artefatos no Rational Unified Process têm representação em UML.
- ♦ Rational Unified Process também inclui as diretrizes para conceitos da UML.

# O processo visto como um produto

- ◆ Entregue em fonte, como um Web site
- ◆ Melhorado continuamente; com atualizações regulares
- ◆ Templates e Ajuda Online propiciam um aprendizado rápido

## Processo Online



## Manuais do Processo



# Estrutura do RUP

---

- ◆ **Organização ao longo do tempo**
  - Estrutura do ciclo de vida: fases e iterações
  - Estabelecimento de Processo: planejando, executando
  - Gerenciamento de atividades, controle do projeto
- ◆ **Organização baseada em conteúdo**
  - Disciplinas, papéis, artefatos e atividades
  - Configuração e melhoria do processo



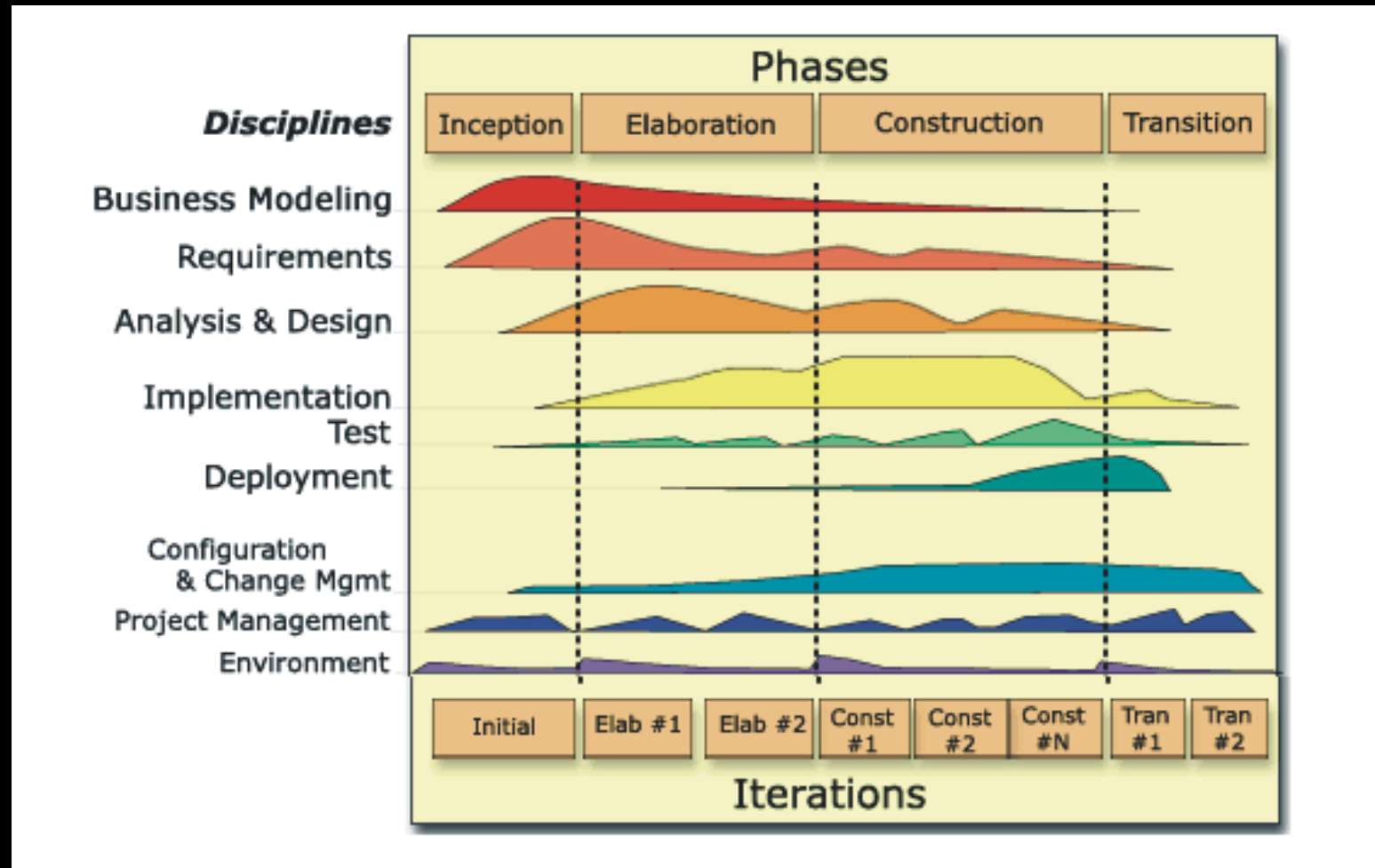
# Conteúdo

---

- ◆ O papel de um processo de desenvolvimento de software
  - ➔ A organização das fases no RUP
- ◆ A organização das disciplinas no RUP

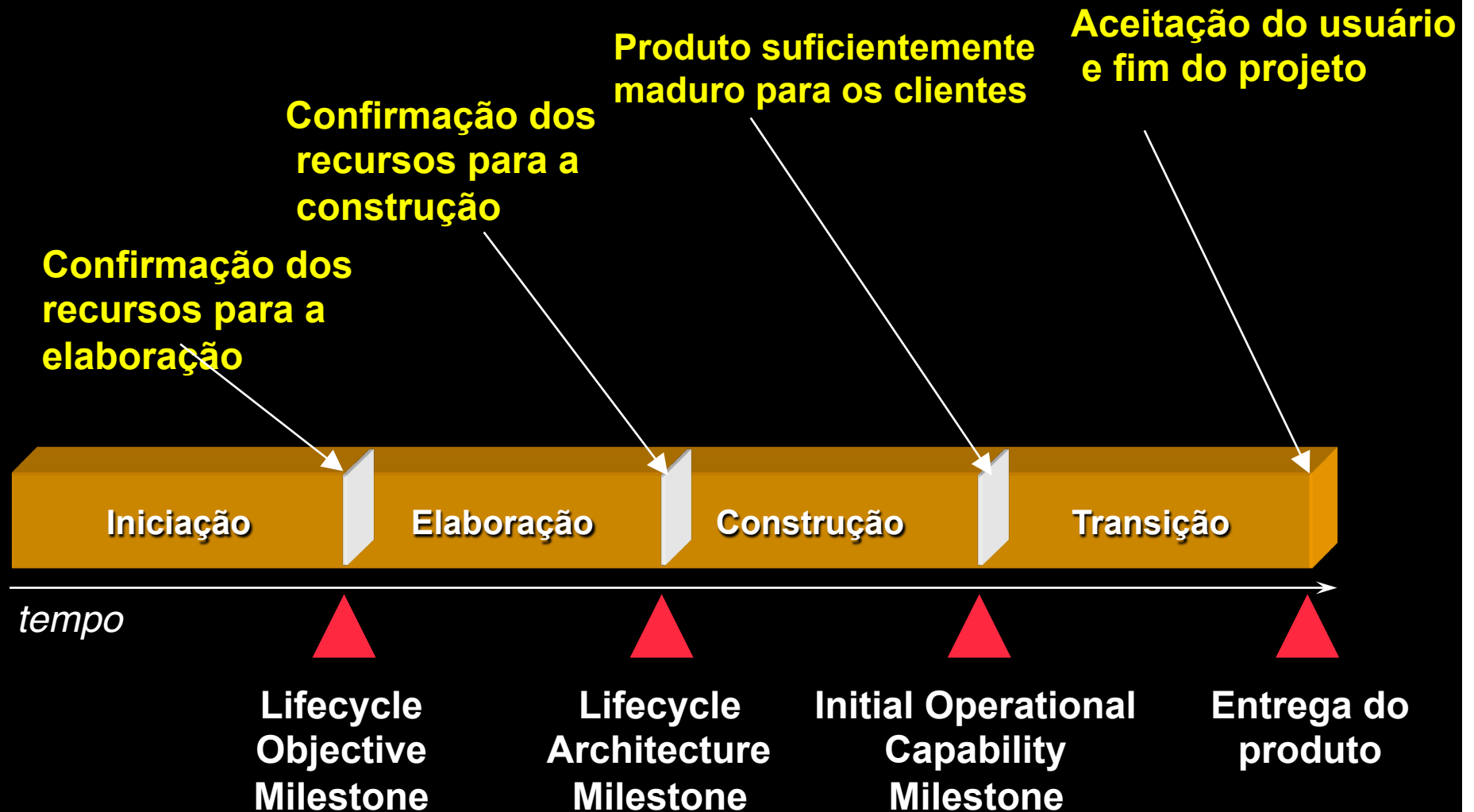
# Organização através do tempo

Tempo →



A organização por fase ajuda a minizar os riscos de alocação de recursos

# Milestones Principais: Pontos de decisões no negócio



# Iniciação: Objetivos

---

- ◆ Preparar o ambiente de suporte para o projeto.
- ◆ Estabelecer o escopo do projeto e as condições limites
- ◆ Determinar os casos de uso e os cenários primários que irão direcionar as principais decisões custo/benefício de design
- ◆ Demonstrar uma arquitetura candidata contra alguns dos cenários primários
- ◆ Estimar o custo e prazo totais
- ◆ Identificar os riscos potenciais (as fontes de imprevisibilidade)

# Iniciação: Critérios de avaliação

---

- Concordância dos principais envolvidos na definição do escopo do projeto e das estimativas de custo/prazo
- Acordo de que o conjunto correto de requisitos foi capturado e que existe entendimento dos mesmos
- Acordo de que as estimativas de custo/prazo, prioridades, riscos, e processo de desenvolvimento estão apropriadas
- Todos os riscos foram identificados e há uma estratégia de redução para cada um deles.

Milestone: Lifecycle Objectives (LCO)

## Elaboração: Objetivos

---

- ◆ Melhorar o ambiente de suporte
- ◆ Definir, validar e criar uma arquitetura baseline o mais rápido possível
- ◆ Baseline da visão
- ◆ Baseline do plano detalhado para a fase de construção
- ◆ Demonstrar que a arquitetura baseline irá suportar a visão, a um custo razoável, em um período razoável de tempo

# Elaboração: Critérios de avaliação

- A Visão de Produto e de requisitos está estável
- Arquitetura está estável.
- Testes chaves e abordagens de avaliação são provadas; os maiores elementos de risco foram encontrados e resolvidos.
- Planos de iteração para a fase de construção estão em detalhes e fidelidade suficientes para que o trabalho proceda, e estão apoiados por estimativas críveis.
- Todos os envolvidos concordam que a visão atual pode ser atingida se o plano atual for executado para desenvolver o sistema completo, no contexto da arquitetura atual.
- Gastos reais de recursos versus gastos previstos são aceitáveis.

Milestone: Lifecycle Architecture (LCA)

## Construção: Objetivos

---

- ◆ Finalizar o produto de software para produção
- ◆ Minimizar os custos de desenvolvimento através da otimização de recursos, evitando retrabalhos e retalhamento desnecessários
- ◆ Obter qualidade adequada o mais rápido possível
- ◆ Obter versões úteis (alfa, beta e outros releases de teste) o mais rápido possível



# Construção: Critérios de avaliação

---

- Esta release de produto é estável e madura o suficiente para ser implantada na comunidade de usuários?
- Todos os envolvidos estão prontos para a transição do produto para o ambiente do usuário?
- As despesas reais com recursos versus despesas planejadas ainda são aceitáveis?

Milestone: Initial Operational Capability (IOC)  
“beta”

## Transição: Objetivos

---

- ◆ Obter a independência do usuário
- ◆ Obter concordância dos envolvidos para que as baselines da implantação sejam completas e consistentes com o critério de avaliação da visão
- ◆ Obter uma baseline do produto final o mais rápido e de maneira mais econômica possível

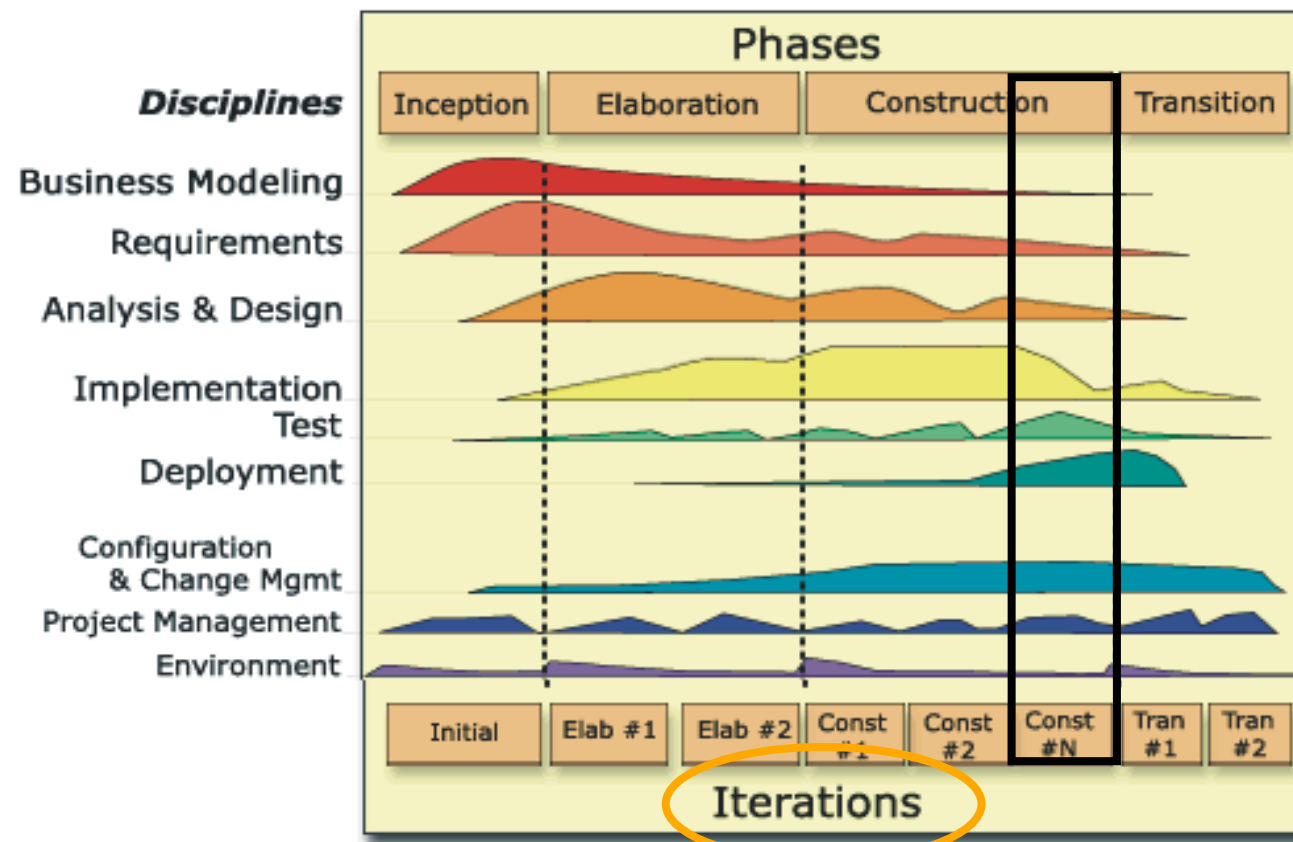
## Transição: Critérios de avaliação

---

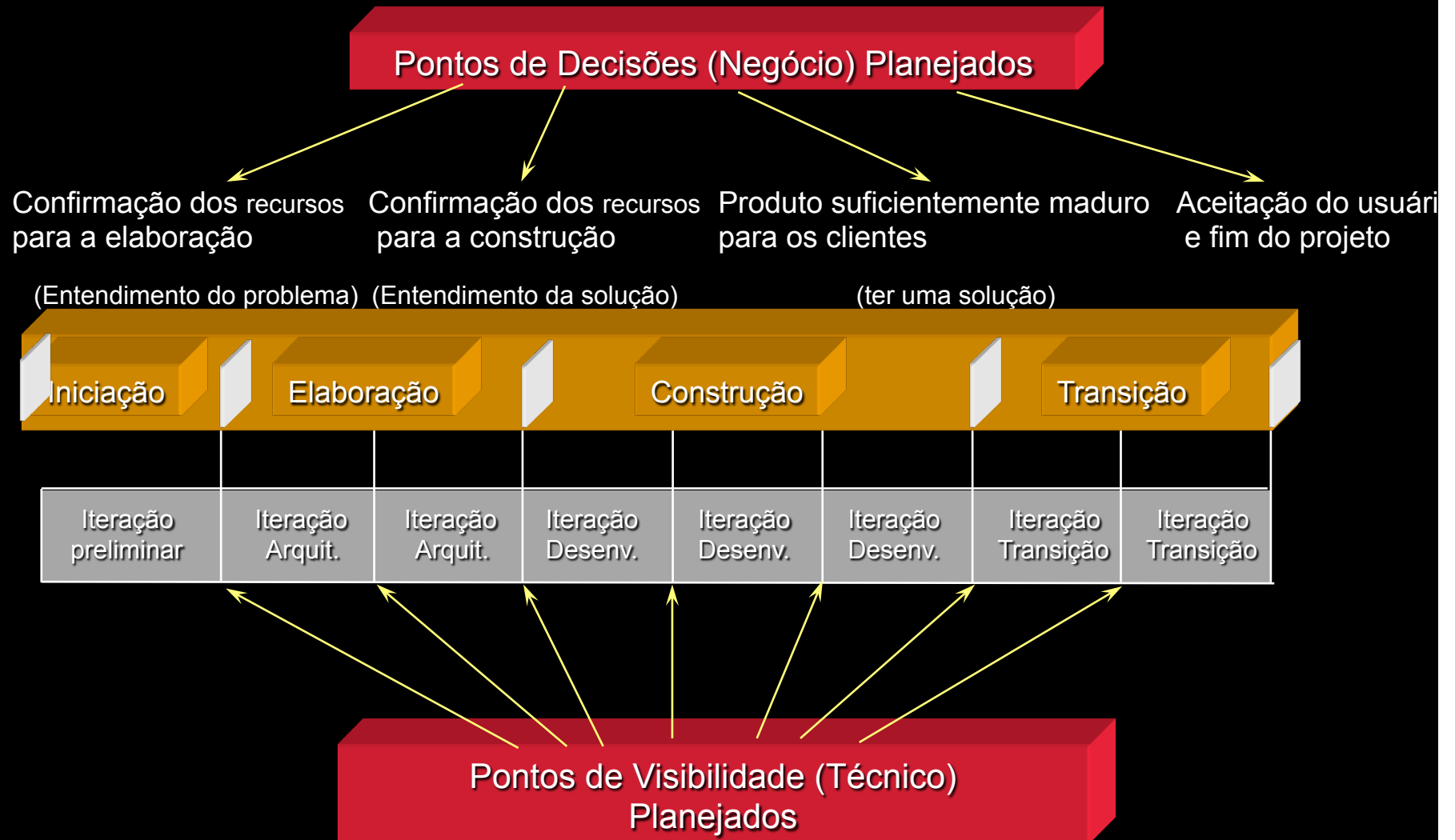
- ◆ O usuário está satisfeito?
- ◆ As despesas reais com recursos versus despesas planejadas ainda são aceitáveis?

Milestone: Product release (“GA”)

# O que é uma iteração?



# Fases e Iterações



# Iteração: duração e quantidade

---

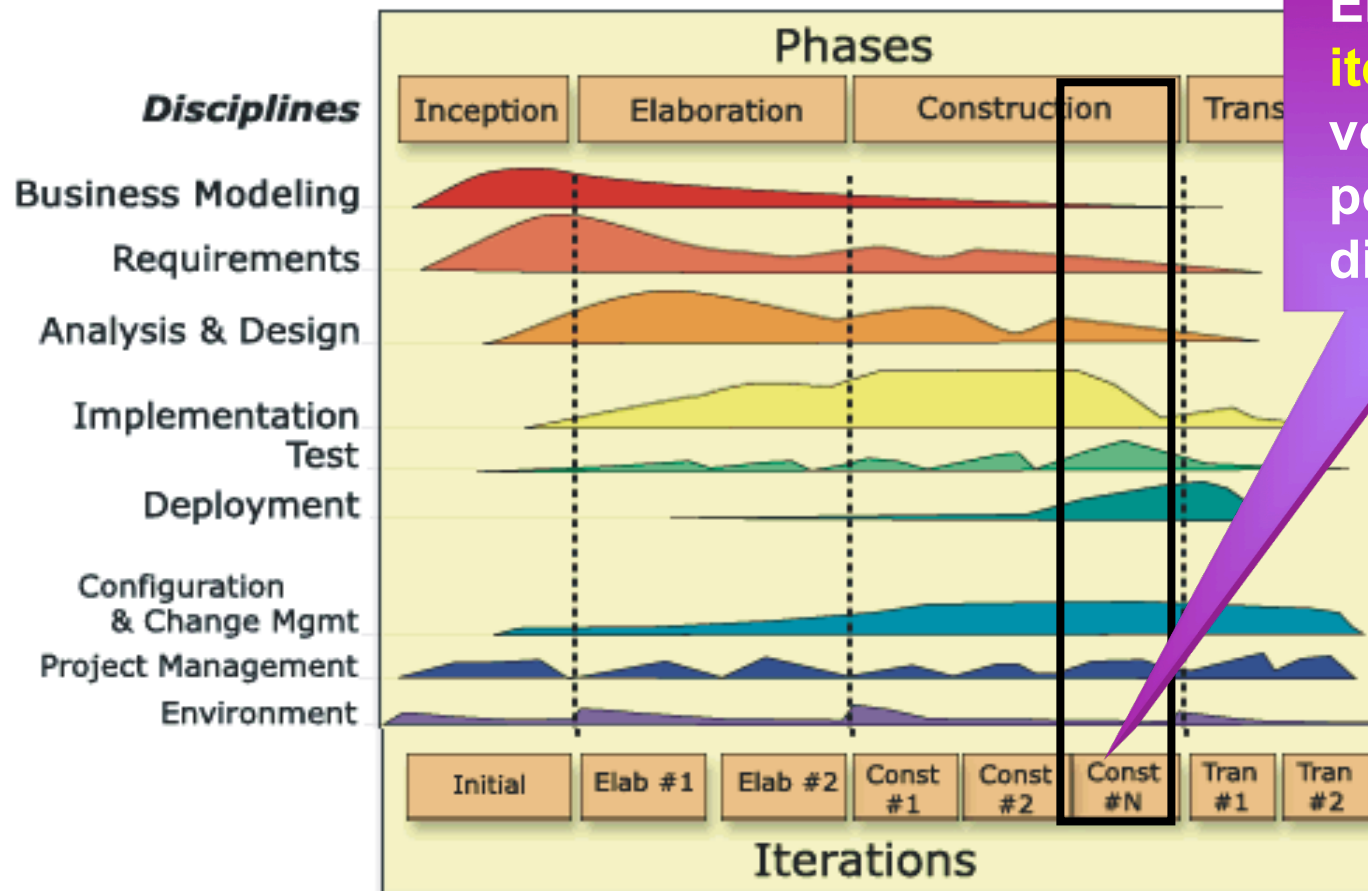
- ♦ O tempo de duração depende de:

- + tamanho da organização
- + tamanho do projeto
- familiaridade com o processo
- complexidade do projeto

- ♦ 6, mais ou menos 3.

- iniciação: 0 .. 1
- elaboração: 1 .. 3
- construção: 1 .. 3
- transição: 1 .. 2

# Uma iteração



Em uma **iteração**, você passa por todas as disciplinas

## Conteúdo do Módulo 2

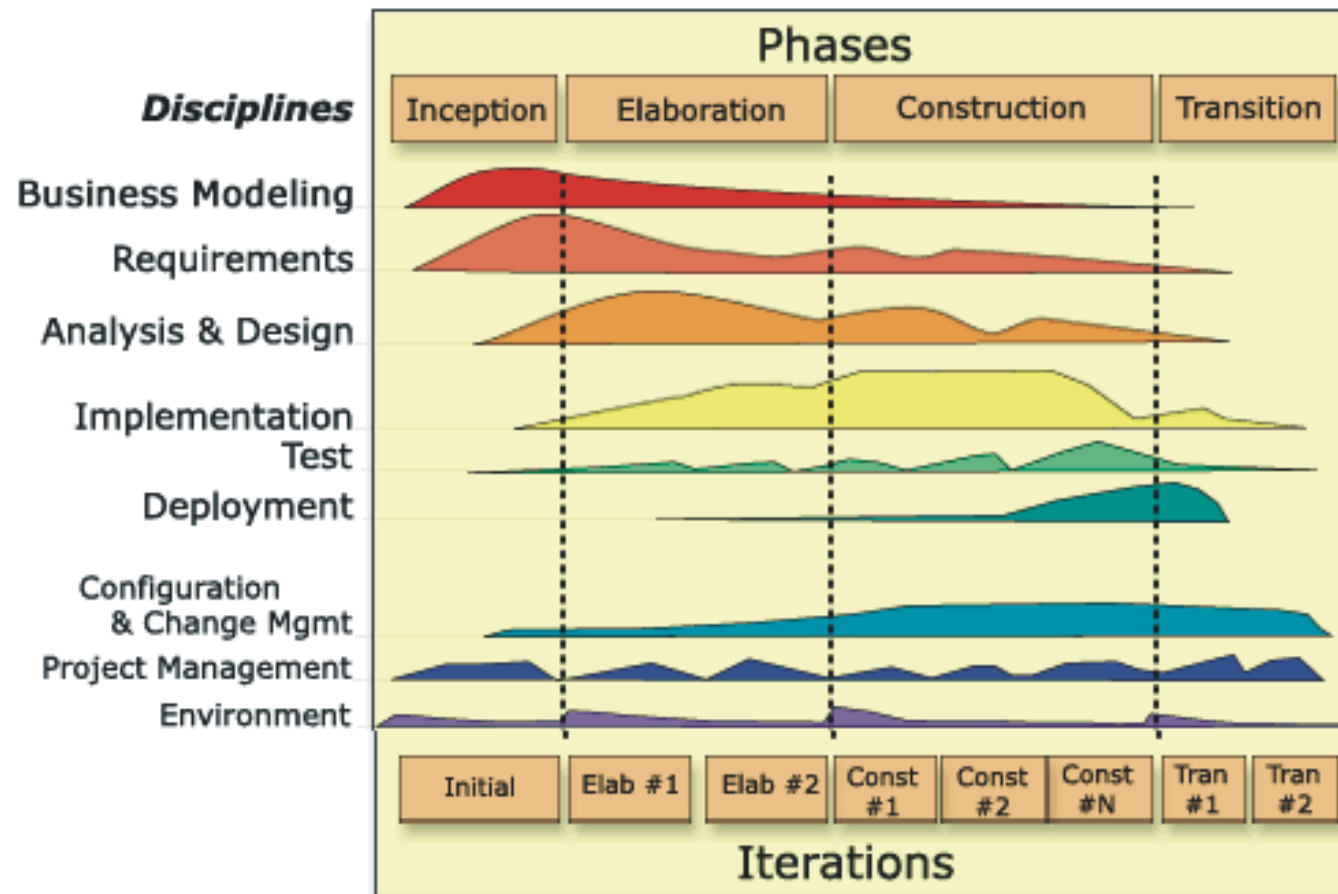
---

- ◆ O papel de um processo de desenvolvimento de software
- ◆ A organização das fases no RUP
- ➔ A organização das disciplinas no RUP

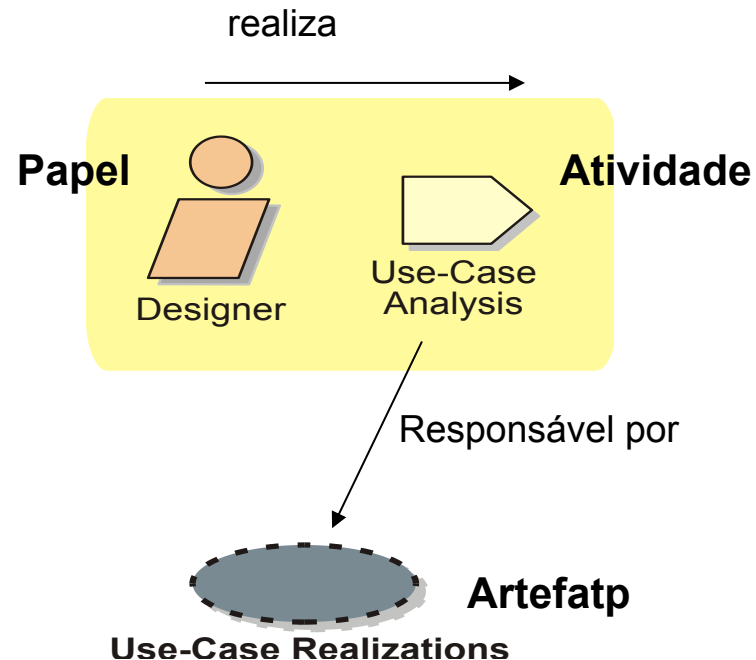


# Organização baseada no conteúdo

Conteúdo

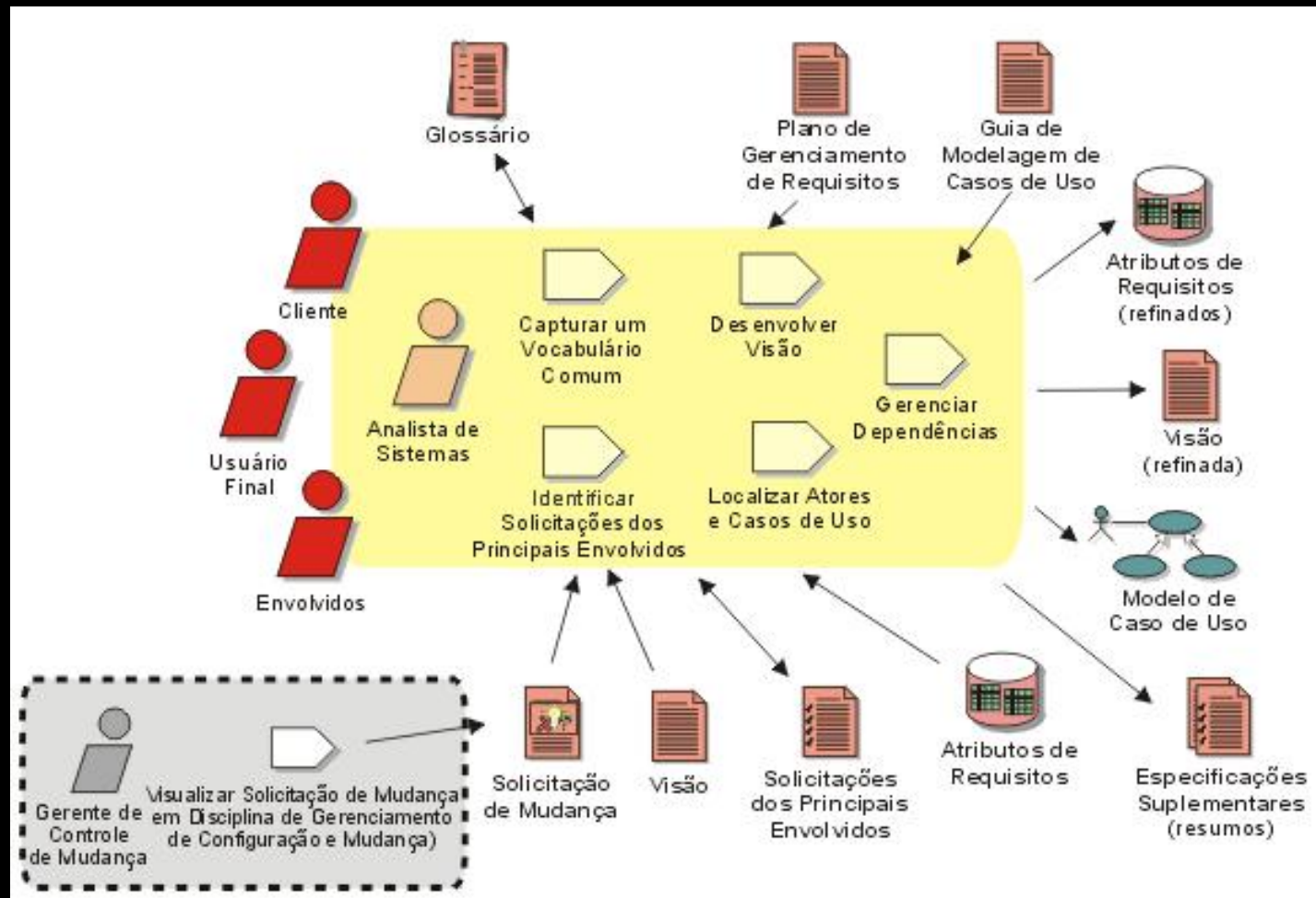


# Elementos chave no RUP: Papel, Atividade e Artefato



# Papéis realizam atividades e produzem artefatos

Exemplo:  
Requisitos->  
Detalhamento do  
Fluxo de Trabalho->  
Compreender as  
Necessidades dos  
Envolvidos



# Elementos chave no RUP: Papel

- ◆ O papel define o comportamento e as responsabilidades de um indivíduo ou de um conjunto de indivíduos que trabalham juntos como uma equipe
- ◆ Os membros da equipe podem desempenhar mais de um papel.



# Papéis são usados para o planejamento de recursos

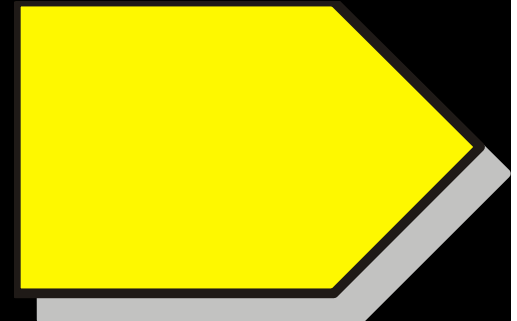


Cada **indivíduo** no projeto é designado para um ou vários **papéis**

## Elementos chave no RUP: Atividade

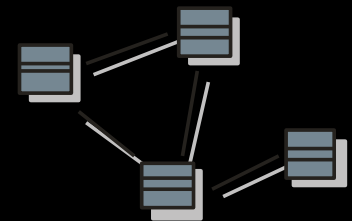
---

- ◆ Uma quantidade de trabalho que um Papel pode ser chamado a realizar e pelo qual é responsável
- ◆ Granularidade: horas, dias
- ◆ Repetida, se necessária, a cada iteração

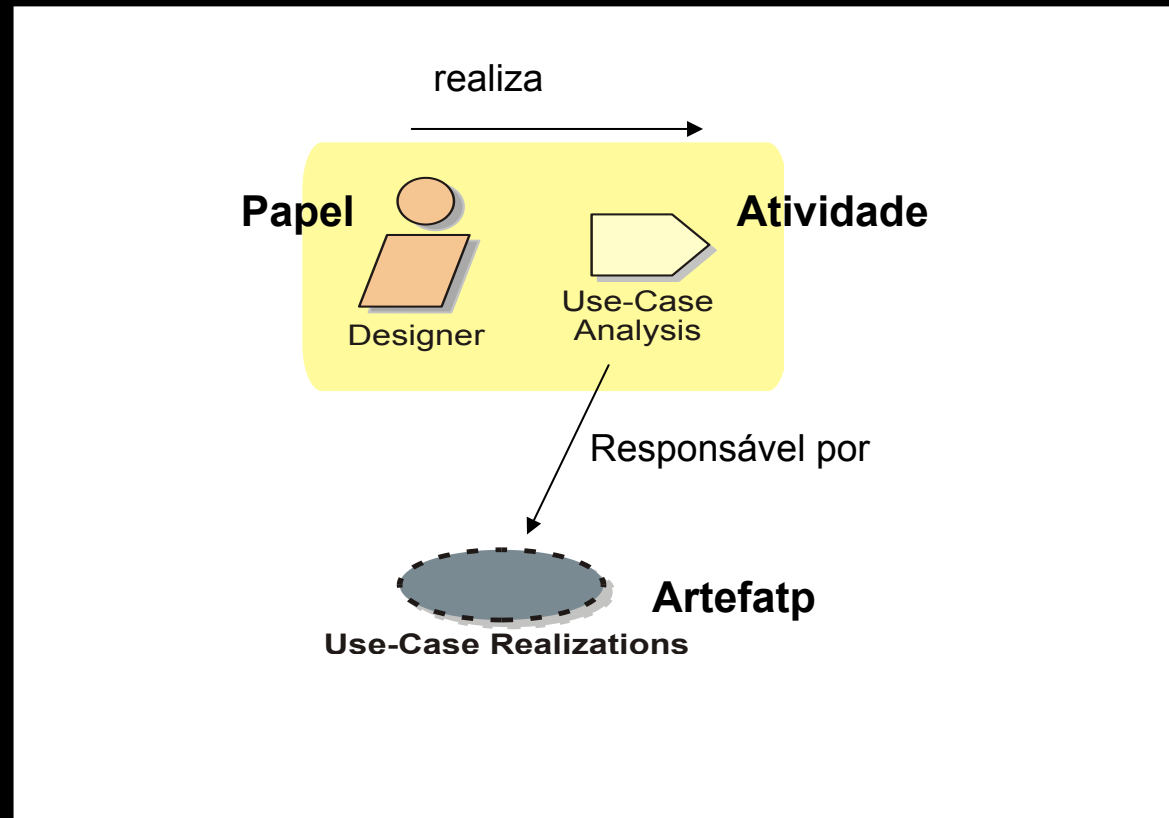


# Elementos chave no RUP: Artefato

- ◆ Um documento ou modelo produzido, evoluído ou usado em um processo
- ◆ Os papéis produzem artefatos ao executarem as atividades
- ◆ Provavelmente estará sujeito ao controle de configuração
- ◆ Pode conter outros artefatos



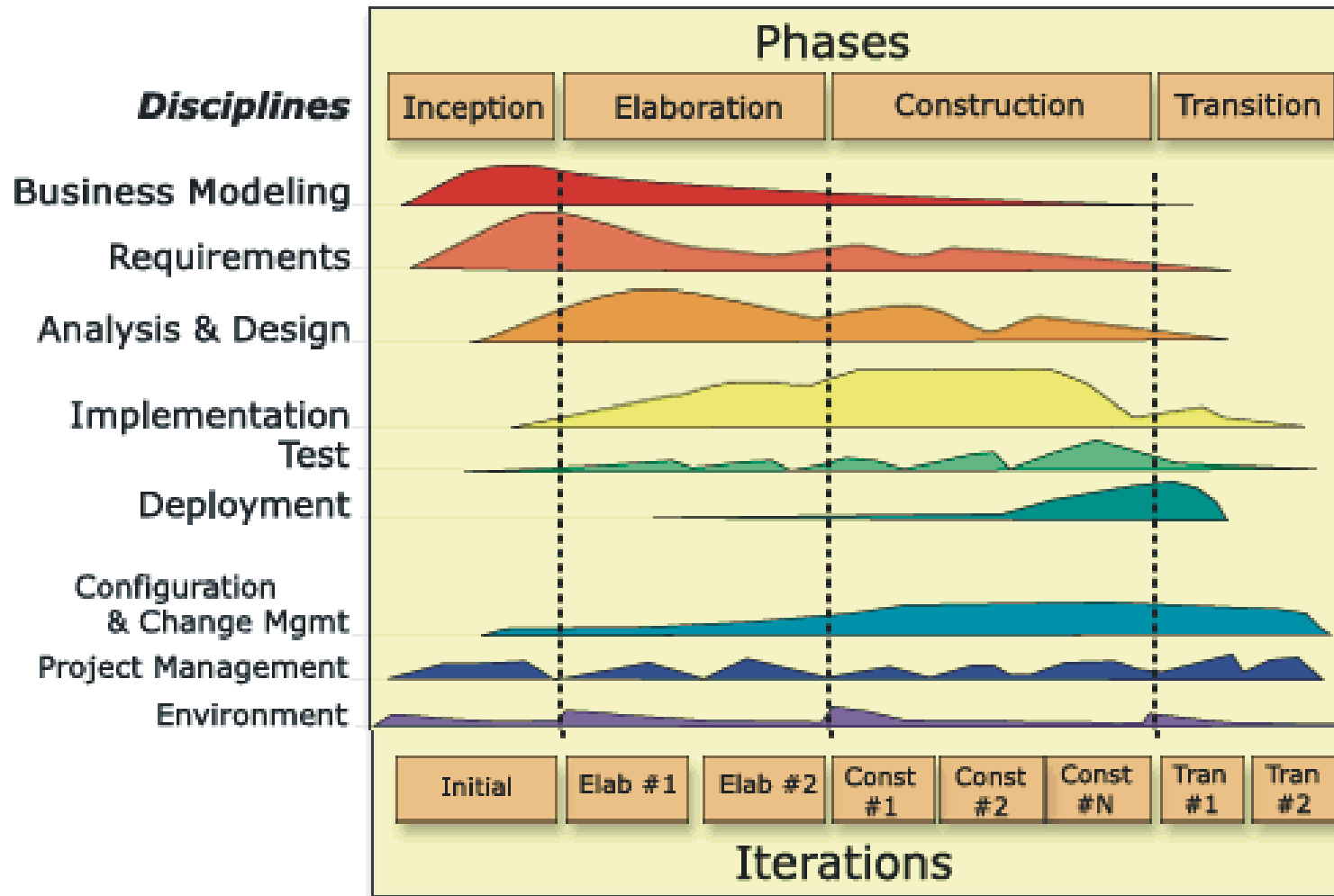
# Elementos chave no RUP: Papel, Atividade e Artefato



Disciplinas também possuem Fluxo de Trabalho e Detalhamento de Fluxo de Trabalho, como veremos adiante.



# Nove Disciplinas



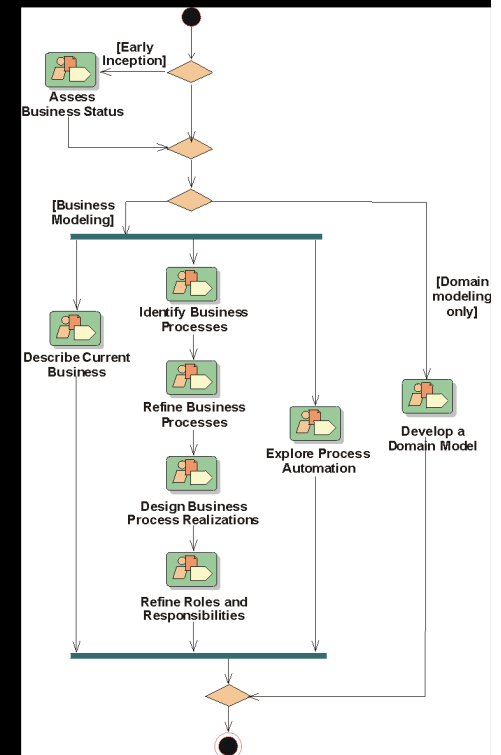
# Elementos de uma disciplina

- ♦ Se você expandir qualquer disciplina, na árvore de navegação do RUP, a seguinte estrutura é mostrada:



# Elementos chave no RUP: Fluxo de Trabalho

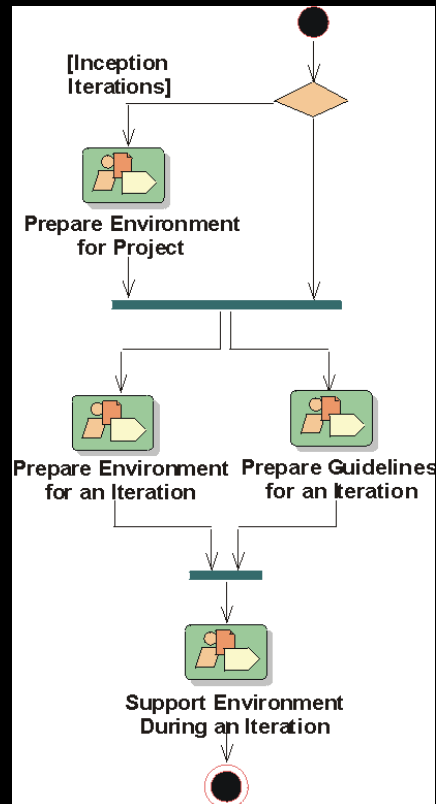
- ◆ Um fluxo condicional de atividades de alto nível (Detalhamento de Fluxo de Trabalho) que produz um resultado de valor significativo.



# Detalhamento do Fluxo de Trabalho

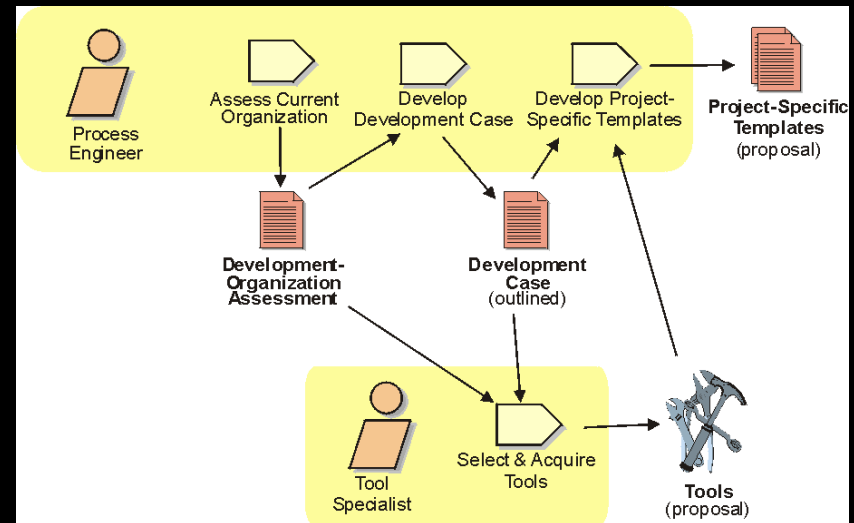
Example:

Ambiente: Fluxo de Trabalho



Example:

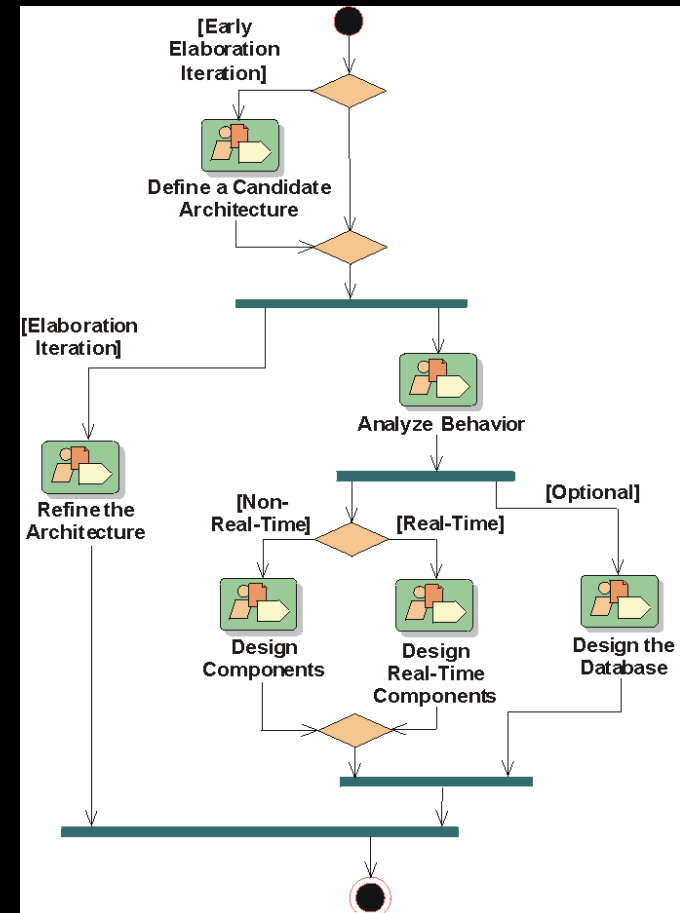
Detalhamento do Fluxo de Trabalho: Preparar o ambiente para o projeto



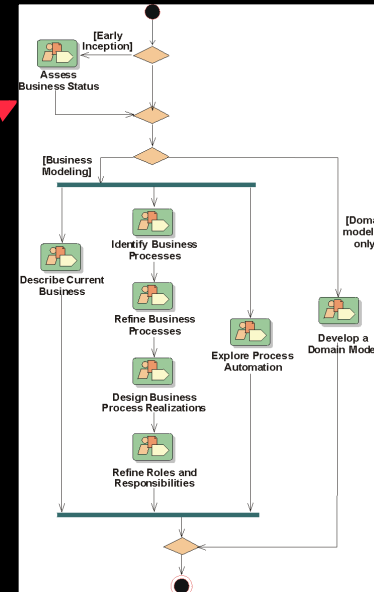
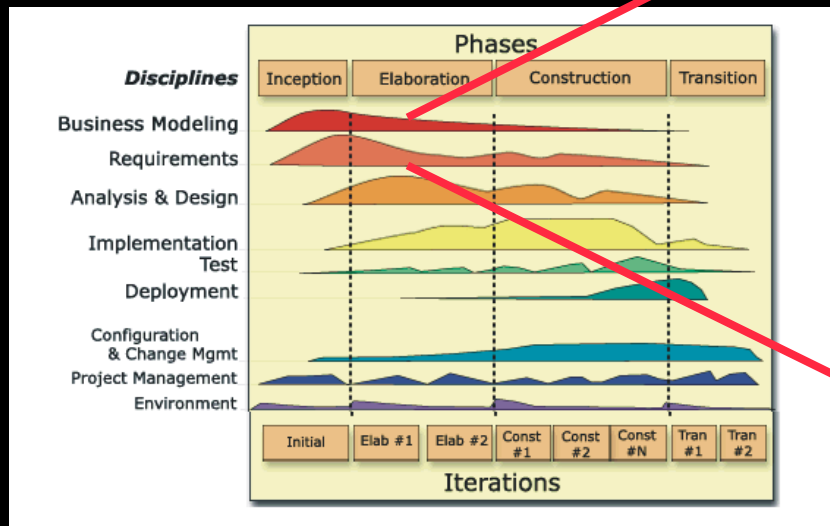
# O Caminho do Fluxo de Trabalho é Adaptável a:

- ◆ **Posição**
  - no Ciclo de vida
  - na Fase
- ◆ **Artefatos sendo produzidos**
- ◆ **Tecnologia**
- ◆ **Objetivos de Iteração**

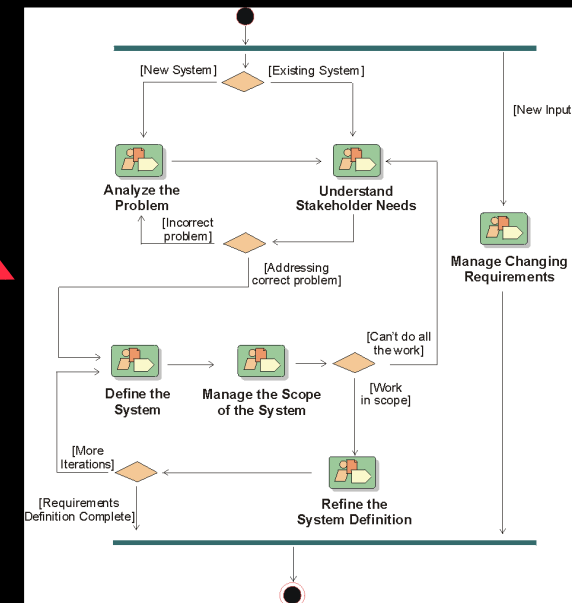
Exemplo:  
Análise & Projeto



# O Detalhamento do Fluxo de Trabalho guia o desenvolvimento iterativo

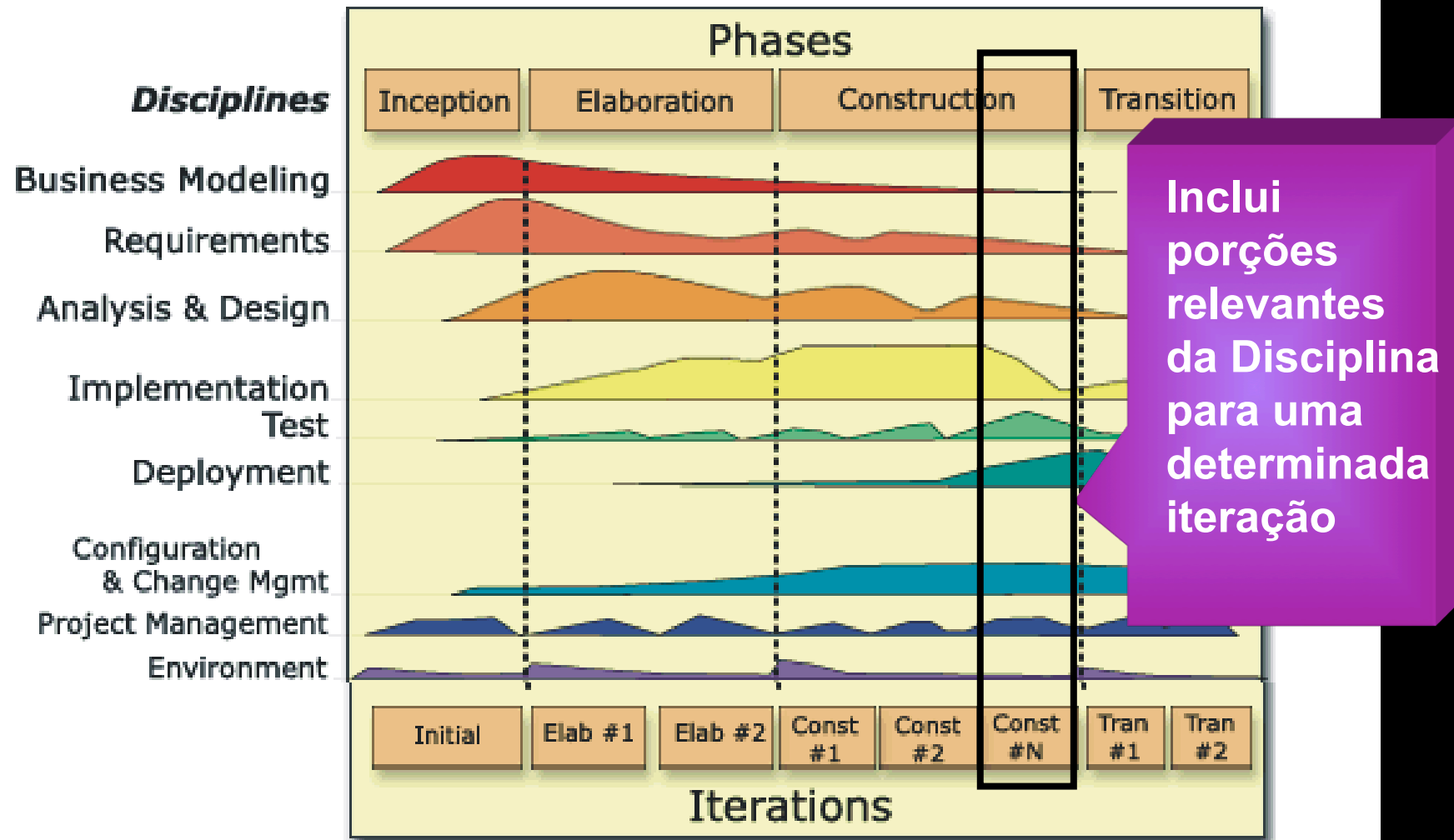


Modelagem de Negócio:  
Detalhamento do Fluxo de Trabalho



Requisitos:  
Detalhamento do Fluxo de Trabalho

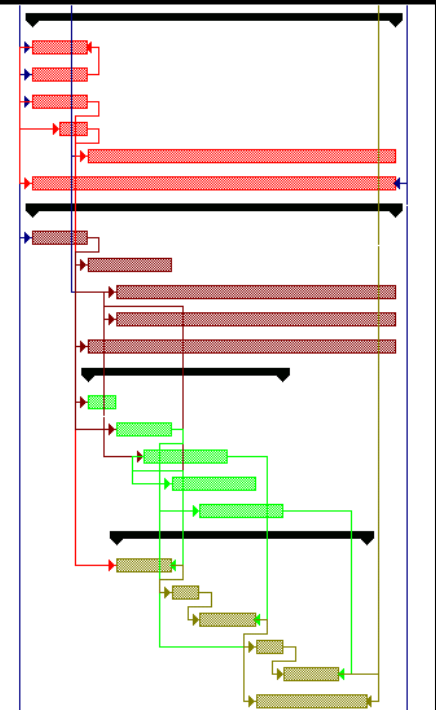
# Plano da Iteração



# Plano da Iteração (cont.)

- ◆ Instância da disciplina
- ◆ Uma por iteração
- ◆ Um plano refinado
- ◆ Expresso em termos de Detalhamento do Fluxo de Trabalho selecionados ou Atividades das disciplinas
- ◆ Mostra os recursos alocados

<b>Requirements</b>
Analyze the Problem
Understand Stakeholder Needs
Define the System
Manage the Scope of the System
Refine the System Definition
Manage Changing Requirements
<b>Analysis and Design</b>
Define a Candidate Architecture
Analyze Behavior
Design Components/Design for Real-Time
Design the Database
Refine the Architecture
<b>Implementation</b>
Structure the Implementation Model
Plan the Integration
Implement Components
Integrate Each Subsystem
Integrate the System
<b>Test</b>
Plan Test
Design Test
Implement Test
Execute Tests in Integration Test Stage
Execute Tests in System Test Stage
Evaluate Test





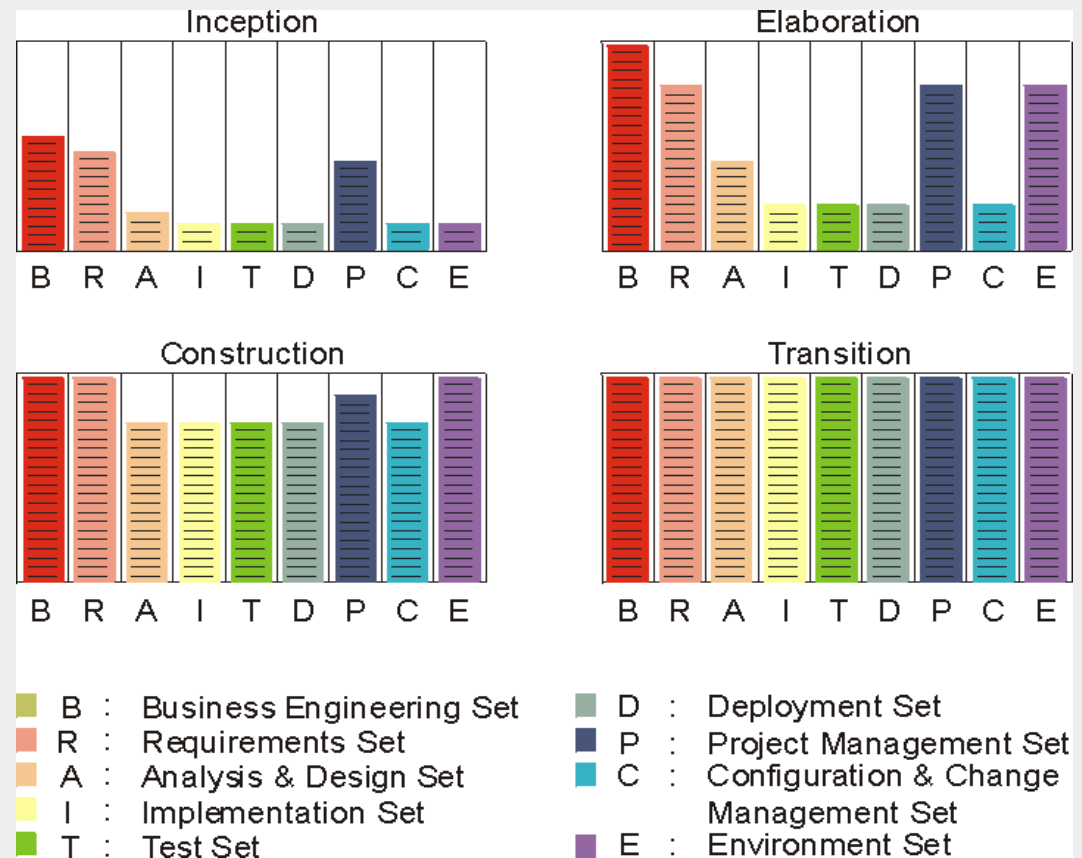
# Exemplo de um plano de iteração

Exemplo:  
Tabela de  
passos para  
uma iteração  
de iniciação

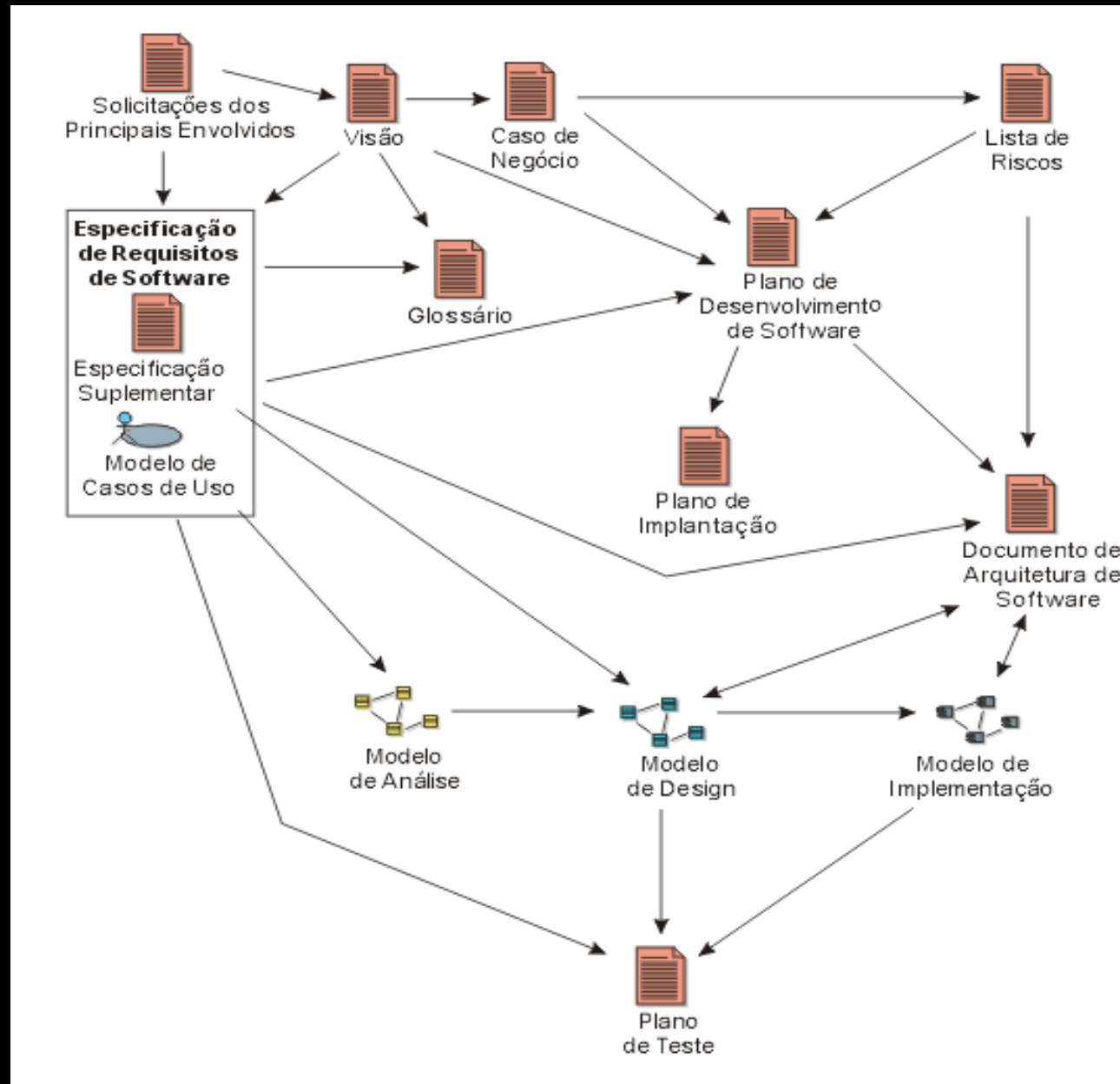
<b>Início: Descrever o plano de iteração, os riscos e os objetivos de arquitetura.</b>	<p>O <a href="#">Artefato: Plano de Iteração</a> para esta iteração foi construído pelo <a href="#">Papel: Gerente do Projeto</a> depois que a iteração anterior foi avaliada, e o escopo e o risco do projeto foram reavaliados. Os critérios de avaliação da arquitetura são descritos pelo <a href="#">Papel: Arquiteto de Software</a> em <a href="#">Artefato: Documento de Arquitetura do Software</a>, levando em consideração os "riscos de arquitetura" que serão amenizados (consulte <a href="#">Artefato: Lista de Riscos</a>). Lembre-se de que uma das metas da Elaboração é estabelecer uma arquitetura sofisticada e executável; o plano para fazê-lo precisa ser desenvolvido na iteração Elaboração inicial.</p>
<b>Ambiente: Preparar ambiente para a iteração.</b>	<p>O <a href="#">Papel: Engenheiro do Processo</a> e o <a href="#">Papel: Especialista em Ferramentas</a> preparam o ambiente para a iteração (consulte <a href="#">Detalhamento do Fluxo de Trabalho: Preparar Ambiente para uma Iteração</a>). Uma informação importante é a avaliação da iteração anterior. O <a href="#">Papel: Engenheiro do Processo</a> conclui o <a href="#">Artefato: Caso de Desenvolvimento</a> e adapta os templates (consulte <a href="#">Artefato: Templates Específicos do Projeto</a>), para estarem prontos para a iteração, adaptando (pelo menos) a disciplina Análise e Design e a disciplina Implementação. O <a href="#">Papel: Especialista em Ferramentas</a> define as ferramentas (consulte <a href="#">Artefato: Ferramentas</a>) que serão utilizadas na iteração. Se necessário, produzirá um <a href="#">Artefato: Guia de Ferramentas</a>. As diretrizes relevantes são desenvolvidas (consulte <a href="#">Detalhamento do Fluxo de Trabalho: Preparar Diretrizes para uma Iteração</a>).</p>
<b>Requisitos: Decidir o que "conduzirá" o desenvolvimento da</b>	<p>O <a href="#">Papel: Arquiteto de Software</a> e o <a href="#">Papel: Gerente de Projeto</a> definem quais casos de uso e/ou cenários devem ser tratados na iteração atual; eles, por sua vez, direcionam o desenvolvimento da</p>

# Evolução dos artefatos com o avanço das fases

- ♦ Com uma abordagem iterativa os artefatos se tornam estáveis com o tempo.



# Resumo dos principais artefatos



# Otimização de Artefatos

---

## ◆ O uso de artefatos no RUP permite:

- Produzir **somente** os artefatos necessários para o seu projeto/ organização
- Mantêm os artefatos nas ferramentas apropriadas (Eclipse, Office, JUDE, Poseidon, etc.)
- Usar relatórios para extrair informações instantâneas de modelos em ferramentas para revisão (scripts, etc.)
- Coloque esforços em artefatos que sejam parte do produto (ex. modelos)

# Elementos Adicionais de Processo

---

## ♦ Disciplina

- Introdução
- **Conceitos**
- Fluxo de Trabalho
- Visão Geral de Atividades
  - **E Mentores de Ferramentas e Diretrizes associados**
- Visão Geral de Artefatos
  - **E Templates e Diretrizes associados**
- Visão Geral de Diretrizes
- **Roteiros**

# Elementos do Processo: Conceito

---

- ◆ Relacionados a uma disciplina
- ◆ Explica as idéias principais
- ◆ Exemplos de conceitos
  - Requisitos
    - Gerenciamento de Requisitos
    - Tipos de Requisitos
    - Rastreabilidade
  - Análise e Projeto
    - Arquitetura do Software
    - Mecanismos de Análise
    - Padrões de Arquitetura Web



# Elementos do Processo: Guias, Normas

---

- ◆ São normas, recomendações, que suportam as atividades e seus passos.
- ◆ Permitem:
  - Descrever técnicas específicas
  - Transformações de um artefato para outro
  - O uso da UML
  - Direcionar o projeto



# Elementos do Processo: Guias, Normas

---

- ◆ São curtas e diretas.
- ◆ Descrevem artefatos bem formados e focam em qualidade.
- ◆ São também usadas para avaliar qualidade dos artefatos.
- ◆ São personalizadas para o projeto.





# Elementos do Processo: Mentores de Ferramentas

---

- ◆ Relacionado a uma atividade relevante
- ◆ Explicam como usar uma ferramenta específica para realizar uma atividade



# Elementos do Processo: Templates

---

- ◆ Anexo a um tipo de documento
- ◆ Artefatos predefinidos:
  - Documentos (Microsoft® Word™, Adobe® Framemaker™)
  - MS Project
  - HTML
- ◆ Personalizado para o processo



# Elementos do Processo: Roteiros

---

Roteiros são utilizados para:

- ◆ Aplicar o processo de propósito geral para resolver tipos específicos de problemas.
- ◆ Descrever as variáveis do processo utilizando fases.
- ◆ Fornecer um mecanismo para estender e adaptar o processo.
- ◆ Destacar algumas características de processo para se conseguir um objetivo em particular.



# Revisão

---

- ◆ Qual é o propósito de um processo de desenvolvimento de software?
- ◆ O que é a organização de fases do RUP?
- ◆ O que é a organização de disciplinas do RUP?
- ◆ Como as iterações estão relacionadas com as fases?
- ◆ Quais são alguns dos principais artefatos?
- ◆ Qual é o objetivo de uma diretriz?