

# Engenharia de Software II

## Aula 4

<http://www.ic.uff.br/~bianca/engsoft2/>

# Modelos Prescritivos de Processo

- Modelo em cascata
- Modelos incrementais
  - Modelo incremental
  - Modelo RAD
- Modelos evolucionários
  - Modelo de prototipagem
  - Modelo espiral
  - Modelo de desenvolvimento concorrente
- Modelos especializados de processo
  - Desenvolvimento baseado em componentes
  - Modelo de métodos formais
  - Desenvolvimento orientado a aspectos

# Comparação

## **Modelo Incremental**

- Atividades fixas do modelo em cascata são usadas em cada incremento.
- Objetiva a elaboração de um produto operacional a cada incremento, que pode ser testado.

## **Modelo Espiral**

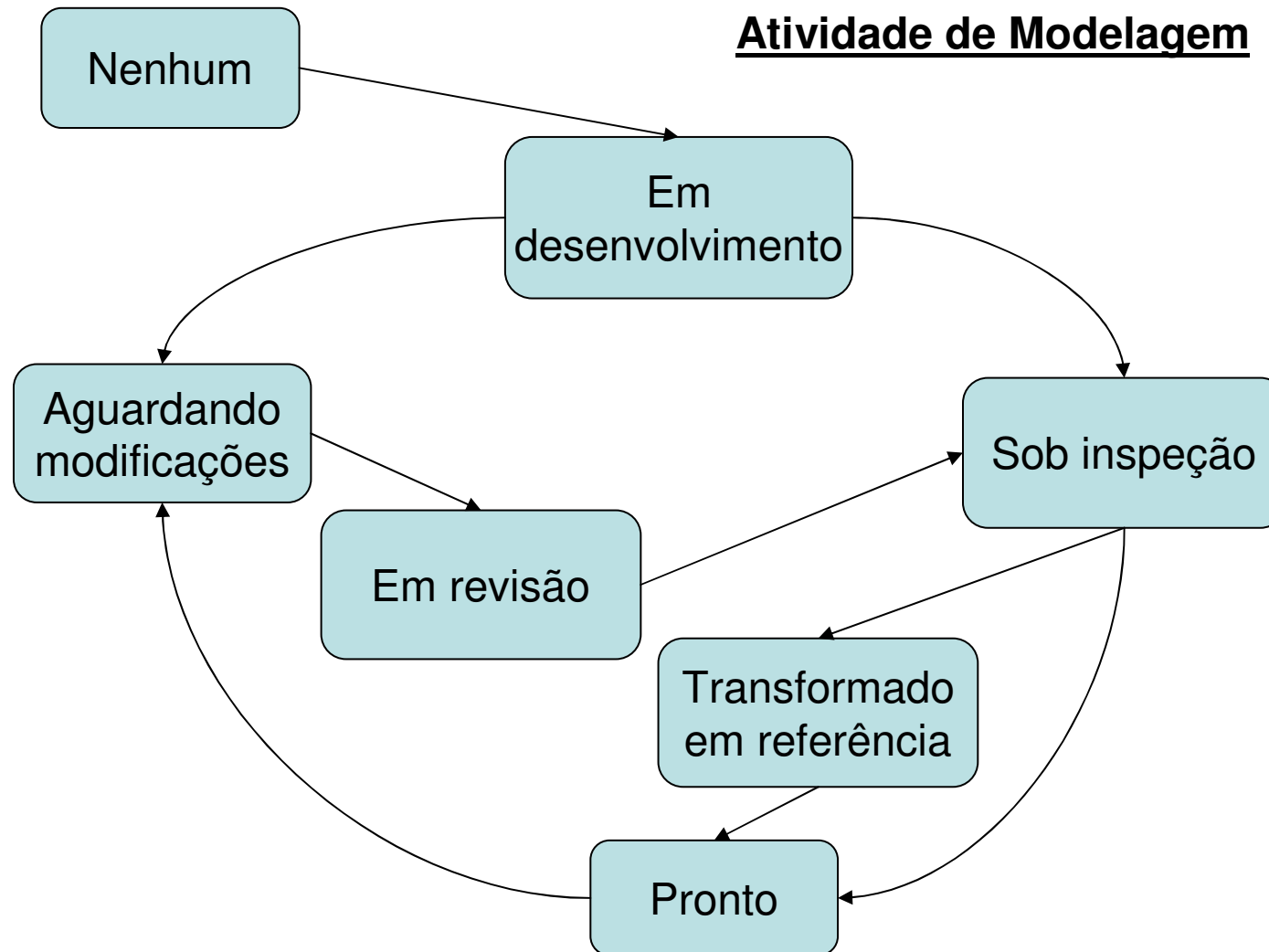
- As atividades não são fixas, cada “loop” se concentra mais em uma determinada atividade.
- A análise de riscos é uma atividade essencial no modelo.

# Modelo de Desenvolvimento Concorrente

- Todas as atividades ocorrem em **paralelo** mas estão em diferentes **estados**.
- O modelo define uma série de eventos que vão disparar **transições** de estado para estado, para cada uma das atividades.
- Em vez de usar uma seqüência como o modelo cascata, ele define uma **rede** de atividades.
- Pode ser aplicado a todo tipo de desenvolvimento de software e fornece uma visão exata de como está o estado do projeto.

# Desenvolvimento Concorrente

## Atividade de Modelagem



# Desenvolvimento Concorrente

- Exemplo: Começo de projeto
  - A atividade de **comunicação** completou sua primeira iteração e está no estado **aguardando modificações**.
  - A atividade de **modelagem** passa do estado **nenhum** para o estado **em desenvolvimento**.
  - Se o cliente requiere mudança nos requisitos, a modelagem passa de **em desenvolvimento** para **aguardando modificações** e a comunicação passa de aguardando modificações para **em revisão**.

# Desenvolvimento Baseado em Componentes

- Compõe aplicações a partir de componentes de software previamente preparados.
- Segue os seguintes passos implantados com uma abordagem evolucionária:
  1. Pesquisa e avaliação de componentes disponíveis para o domínio em questão.
  2. Considerações sobre a integração de componentes.
  3. Projeto de arquitetura de software.
  4. Integração dos componentes à arquitetura.
  5. Testes para garantir a funcionalidade adequada.

# Vantagens do desenvolvimento baseado em componentes

- Leva ao **reuso** de software, que segundo estudos tem como consequências:
  - Redução significativa do prazo de desenvolvimento.
  - Redução significativa no custo do projeto.
  - Aumento do índice de produtividade.
- Em que situações o desenvolvimento baseado em componentes **não** é adequado?
  - Aquelas em que não existam componentes padrão disponíveis ou em que não se queira pagar pelos componentes.



# Modelo de Métodos Formais

- Métodos formais permitem ao engenheiro de software especificar, desenvolver e verificar um sistema aplicando uma **rigorosa notação matemática**.
  - Uma variante chamada **engenharia de software sala limpa** é aplicada por algumas organizações.

# Vantagens dos métodos formais

1. **Elimina** muitos problemas encontrados nos outros modelos:
  - ambigüidade
  - incompletitude
  - inconsistência
2. Servem de base para a **verificação** de programas, oferecendo a promessa de um software livre de defeitos.
3. Adequado para softwares **críticos** (por exemplo, de aeronaves e dispositivos médicos).

# Desvantagens dos métodos formais

1. O desenvolvimento de modelos formais é atualmente muito **lento** e dispendioso.
2. Exige **treinamento** extensivo para dar aos desenvolvedores o preparo necessário.
3. É difícil usar os modelos como um mecanismo de **comunicação** com a maioria dos clientes.

# Desenvolvimento Orientado a Aspectos

- É um paradigma **novo** de engenharia de software que fornece mecanismos para definir, especificar, projetar e construir **aspectos**.
- Aspectos=preocupações do cliente que permeiam **diversos níveis** do sistema, incluindo:
  - Propriedades de alto nível (ex: segurança, tolerância a falha).
  - Funções (ex: aplicação de regras de negócio).
  - Sistêmicas (ex: sincronização e gestão de memória).
- Um processo orientado a aspectos ainda não foi totalmente desenvolvido, mas deve adotar características do modelo espiral e do modelo concorrente.

# O Processo Unificado

- É uma tentativa de **unir** os melhores recursos e características dos modelos convencionais.
- Reconhece a importância da comunicação com o cliente e dos **casos de uso** para descrever a visão do cliente
- Utiliza a **UML** como a notação para modelagem e análise de projeto.
- Sugere um fluxo de processo que é **iterativo** e **incremental**.
- Também conhecido como RUP (de Rational Unified Process) – a Rational construiu ferramentas de apoio ao processo unificado.

# Histórico do Processo Unificado

- **Década de 1980:** popularização dos métodos de programação orientada a objeto (OO) levando a métodos variados de análise e projeto OO.
- **Início da década de 1990:** Rumbaugh, Booch e Jacobson começaram a trabalhar em um “método unificado”, que resultou na UML.
  - A UML tornou-se uma norma industrial.
  - A Rational e outros vendedores desenvolveram ferramentas UML.
- **Final da década de 1990:** Jacobson, Rumbaugh e Booch desenvolvem o Processo Unificado, um arcabouço para engenharia de software OO.
- Hoje em dia o PU e a UML são amplamente usados em projetos OO de todas as naturezas.

# Fases do Processo Unificado

1. **Concepção:** abrange atividades de comunicação com o cliente e de planejamento.
  - Requisitos de negócio usando casos de uso preliminares.
  - Arquitetura geral do sistema com os principais subsistemas e funções.
  - Planejamento com recursos, riscos e cronogramas.

# Fases do Processo Unificado

- 2. Elaboração:** abrange as atividades de comunicação com o cliente, planejamento e modelagem.
- Refina e expande os casos de uso preliminares.
  - Expande a representação arquitetural para incluir cinco visões diferentes:
    - O modelo de casos de uso.
    - O modelo de análise.
    - O modelo de projeto.
    - O modelo de implementação.
    - O modelo de implantação.
  - O plano é revisto e pode ser modificado.



# Fases do Processo Unificado

- 3. Construção:** idêntica a atividade de construção no processo genérico.
- Usa o modelo arquitetural como entrada.
  - Desenvolve ou adquire e integra componentes de software.
  - Torna cada caso de uso operacional.
  - Modelos de análise e projeto são completados.
  - Testes são elaborados e executados.

# Fases do Processo Unificado

## 4. **Transição:** abrange atividades de construção e implantação.

- O software é dado aos usuários finais para testes beta e relatórios de *feedback* que podem levar a modificações.
- Informações de apoio necessárias são criadas (manuais e procedimentos de instalação).
- Na conclusão dessa fase tem-se uma versão utilizável do software.

# Fases do Processo Unificado

## 5. **Produção:** abrange atividades de implantação.

- O uso do software é monitorado.
- É fornecido suporte para o ambiente de operação.
- Os relatórios de defeito e solicitações são recebidos e avaliados.

# Processo Unificado

- É um processo **incremental**.
  - Enquanto acontecem as fases de construção, transição e produção, já pode ser iniciado o incremento seguinte.
- Um **fluxo de trabalho** de engenharia de software é distribuído ao longo de todas as fases do PU.
  - Identifica as tarefas exigidas para realizar uma ação importante de engenharia de software.

# Principais Produtos de Trabalho do Processo Unificado

- **Concepção:**
  - Documento de visão
  - Modelo inicial de caso de uso
  - Glossário inicial do projeto
  - Caso de negócio inicial
  - Avaliação inicial de risco
  - Plano de projeto
  - Modelo de negócio
  - Um ou mais protótipos

# Principais Produtos de Trabalho do Processo Unificado

- **Elaboração:**

- Modelo de caso de uso
- Requisitos suplementares
- Modelo de análise
- Descrição da arquitetura do software
- Protótipo arquitetural executável
- Modelo de projeto preliminar
- Lista de risco revisada
- Plano de projeto incluindo planos de iteração, fluxos de trabalho adaptados, marcos, produtos técnicos de trabalho
- Manual preliminar do usuário.

# Principais Produtos de Trabalho do Processo Unificado

- **Construção:**
  - Modelo de projeto
  - Componentes de software
  - Incremento integrado de software
  - Plano e procedimento de teste
  - Caso de teste
  - Documentação de apoio
    - Manuais do usuário
    - Manuais de instalação
    - Descrição do incremento atual

# Principais Produtos de Trabalho do Processo Unificado

- **Transição:**
  - Incremento de software entregue
  - Relatório de teste beta
  - Realimentação geral do usuário