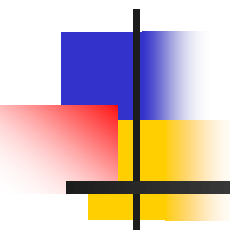


# Processo de Software - Revisão



---

**Universidade Federal do Maranhão - UFMA**  
**Departamento de Informática**  
**Processo de Desenvolvimento de Software**  
Profª.MSc Simara Rocha  
[simararocha@gmail.com/simara@deinf.ufma.br](mailto:simararocha@gmail.com/simara@deinf.ufma.br)

Referências: Pressman, R. S. Engenharia de Software, McGraw-Hill, 6ª. Edição, 2006  
Sommerville, I. Engenharia de Software, 8ª edição, 2007.



# Sumário

---

- Introdução
- Conceitos Básicos
- Processo de Software
- Modelos de processo de Software e seus paradigmas
- Perguntas Comuns



# Conceitos Básicos

---

- Métodos
  - Métodos: proporcionam os detalhes de “como fazer” para construir o software.
  - Envolvem um amplo conjunto de tarefas
  - Um método de ES é uma aproximação estruturada para o desenvolvimento de software
  - Objetivo: Produção de software de alta qualidade e com bom custo-benefício.



# Conceitos Básicos

---

- Métodos devem incluir os seguintes componentes:
  - Descrição gráficas.
    - Descrições dos modelos do sistema que deverão ser desenvolvidos e da notação usada para os definir.
    - Ex. Modelos de objetos, fluxos de dados etc.



# Conceitos Básicos

---

- Métodos devem incluir os seguintes componentes(cont.):
  - Regras
    - Restrições que se aplicam a modelos de sistema.
    - Ex.Cada entidade deve ter um único nome.



# Conceitos Básicos

---

- Métodos devem incluir os seguintes componentes (cont.):
  - Recomendações
    - Conselho em prática de projeto.
    - Ex. Nenhum objeto deve ter mais que sete sub-objetos.



# Conceitos Básicos

---

- Métodos devem incluir os seguintes componentes (cont.):
  - Diretrizes de processo
    - Descrição das atividades que podem ser seguidas.
    - Ex. Atributos de objetos devem ser documentados.



# Conceitos Básicos

---

- Ferramentas: fornecem suporte automatizado ou semi aos métodos.
  - Existem atualmente ferramentas para sustentar cada um dos métodos
  - Quando as ferramentas são integradas é estabelecido um sistema de suporte ao desenvolvimento de software chamado *CASE - Computer Aided Software Engineering*





# Conceitos Básicos

---

- Procedimentos: constituem o elo de ligação que mantém juntos os métodos e as ferramentas para desenvolvimento do software.
  - Seqüência em que os métodos serão aplicados
  - Produtos (*deliverables*) que se exige que sejam entregues
  - Controles que ajudam assegurar a qualidade e coordenar as alterações
  - Marcos de referência que possibilitam administrar o progresso do software



# Processos de Software

---

- Um **processo de software** é um método para desenvolver ou produzir software.
- A pesquisa em **processo de software** lida com métodos e tecnologias estimativas, suporte e melhoria das atividades de desenvolvimento de software.
- Define quem faz *o que, quando e como.*



# Processo de Software

---

- Modelagem: é uma técnica de engenharia aprovada e bem aceita
  - modelos de arquitetura de casas e de grandes prédios
  - modelos matemáticos a fim de analisar os efeitos de ventos e tremores de terra --> causas
- O que é um MODELO?



# Modelos

---

- É uma simplificação da realidade.
  - Planos de detalhes, podendo ser:
    - estruturais (organização do sistema)
    - comportamentais (dinâmica do sistema)

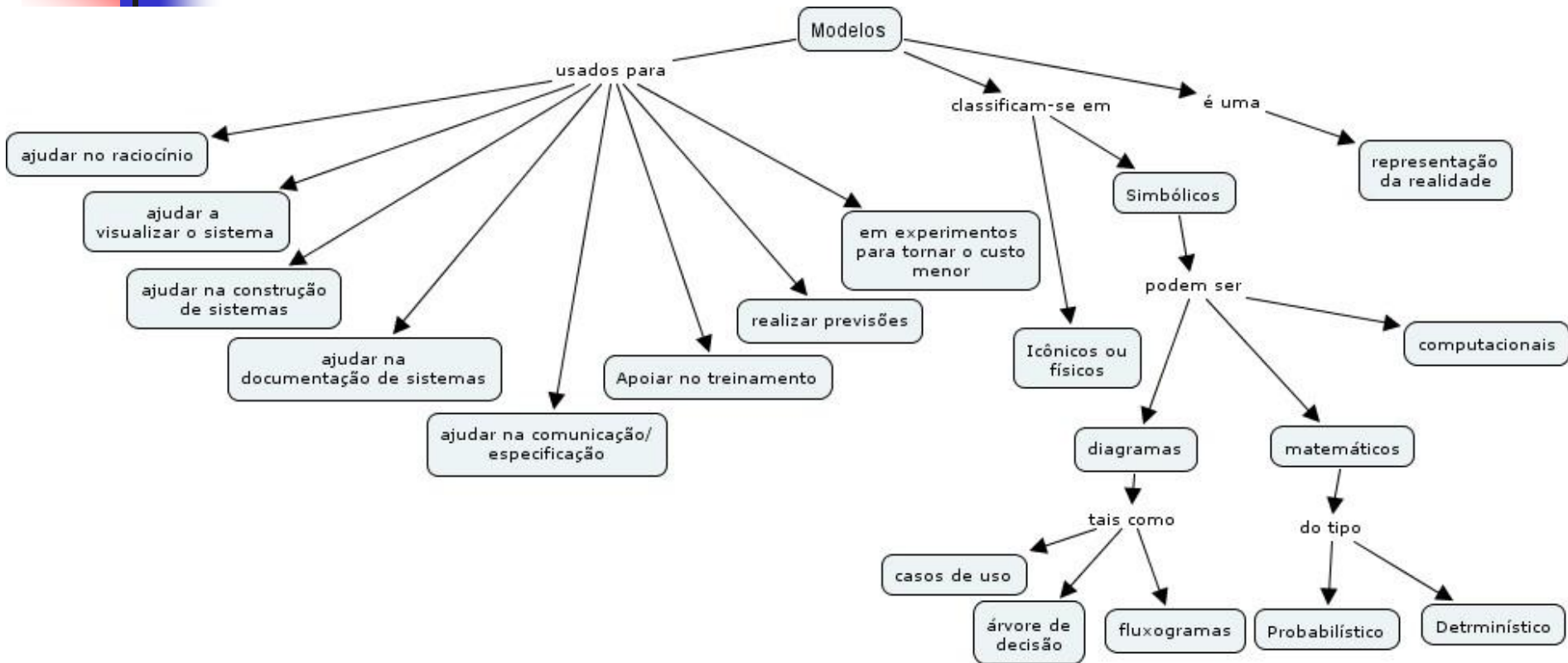


# Modelos

---

- Modelos são construídos para permitir um melhor entendimento sobre o sistema que está sendo construído.
  - especificar a estrutura e comportamento
  - guia para construção do sistema
  - documentam as decisões tomadas
- Nenhum modelo único é suficiente.
  - conjunto de modelos independentes

# Modelos





# Objetivos da Modelagem

---

- Auxiliar no processo de produção → produtos de alta qualidade, produzidos mais rapidamente e a um custo cada vez menor.
- Atributos: abstração, visibilidade, especificação, construção, confiabilidade, manutenibilidade, segurança e documentação.



# Objetivos da Modelagem

---

- Abstração
  - Melhor entendimento e maior compreensão
- Visualização
  - Visualização antecipada antes da implementação
  - Visões complementares do software





# Objetivos da Modelagem

---

- Especificação
  - descrição precisa do que deve ser feito
- Construção
  - geração automática com ferramentas baseadas em modelos
- Documentação
  - comunicação entre equipes nas diferentes fases do ciclo de vida



# Modelo X Processo

---

- Um modelo é algo teórico, um conjunto de possíveis ações.
- O processo deve determinar ações práticas a serem realizadas pela equipe como prazos definidos e métricas para se avaliar como elas estão sendo realizadas

Modelo + Planejamento = Processo



# Modelos de processo de software

---

- Um conjunto de atividades fundamentais exigida para desenvolver um sistema de software
  - Especificação.
  - Projeto e implementação.
  - Validação.
  - Evolução.



# Modelos de processo de software

---

- Um modelo de processo de software é uma representação abstrata de um processo.
- Representa uma descrição de um processo a partir de uma perspectiva particular.



# Modelos de processo de software

---

- Uma representação simplificada de um processo de software, apresentada de uma perspectiva específica
- Exemplos de modelos de processo são:
  - Workflow - sucessão de atividades
  - Fluxo de Dados - fluxo de informação
  - Papel/ação – representa os papéis das pessoas e as atividades pelas quais elas são responsáveis.



# Modelos de processo de software - Paradigmas

---

- Uma estratégia de desenvolvimento que englobe processos, métodos e ferramentas, e as fases de desenvolvimento...
- Exemplo:
  - Modelo Seqüencial (ciclo de vida clássico), Modelo de Prototipação, Modelos Evolutivos (Modelo Incremental e Espiral), Modelo de Métodos Formais, Orientado a reuso, etc.



# Modelo em Cascata

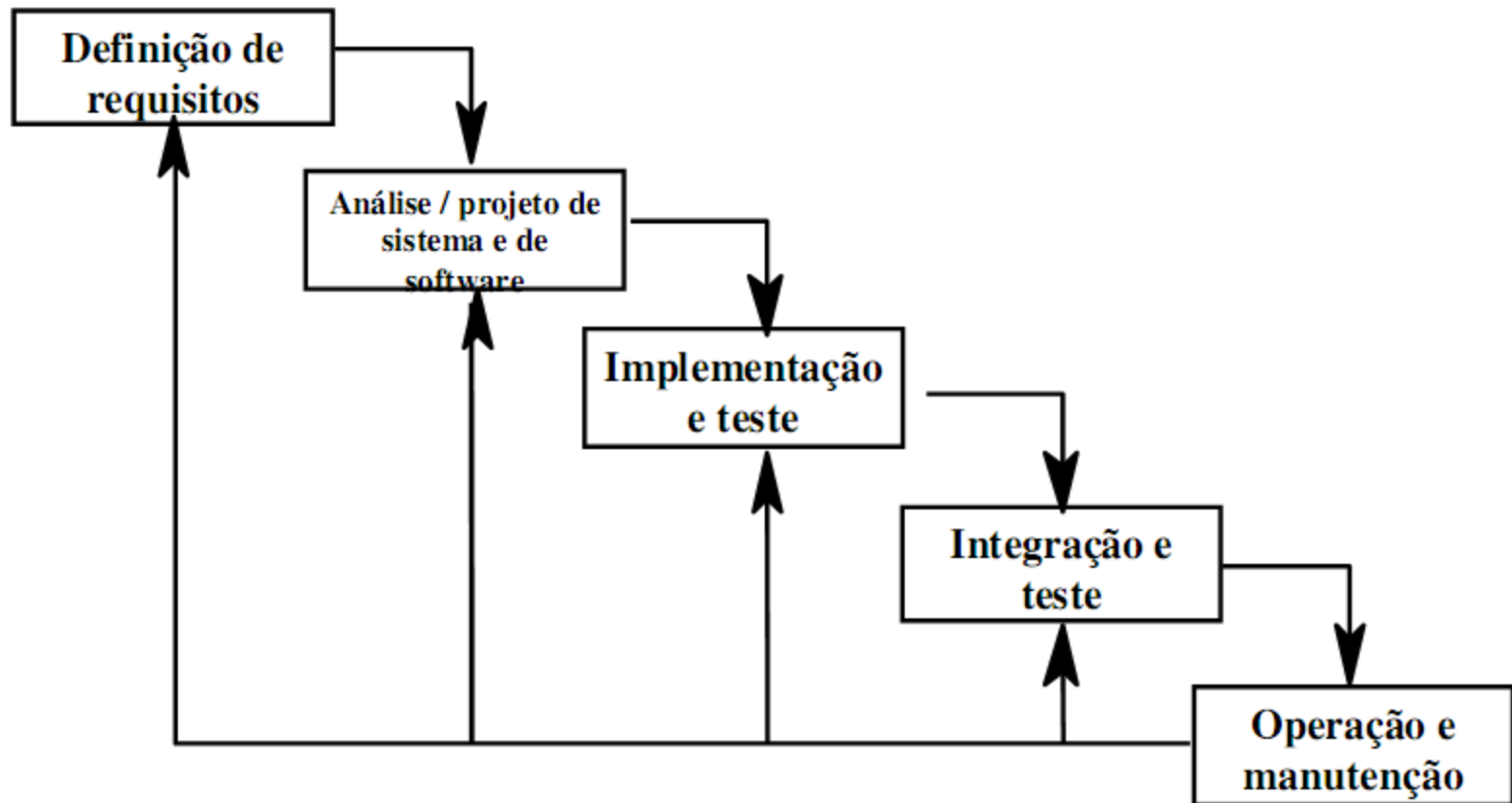
---

- Método sistemático e sequencial
- O resultado de uma fase se constitui na entrada da outra.
- Cada fase é estruturada como um conjunto de atividades que podem ser executadas por pessoas diferentes, simultaneamente.



# Modelo em Cascata

---







# Modelo Evolutivo

---

- Abordagem baseada na idéia de desenvolver uma implementação inicial, expor o resultado ao comentário do usuário e fazer seu aprimoramento por meio de muitas versões.
- As atividade de desenvolvimento e validação são desempenhadas paralelamente, com um rápido feedback entre elas.



# Modelo Evolutivo

---

- Tipos de desenvolvimento Evolutivo:
  - Desenvolvimento Exploratório: trabalha-se junto com o cliente, a fim de explorar seus requisitos e entregar um sistema final.
    - o desenvolvimento se inicia com as partes do sistema que são mais bem compreendidas.
    - o sistema evolui com o acréscimo de novas características à medida que elas são propostas pelo cliente.



# Modelo Evolutivo

---

- Tipos de desenvolvimento Evolutivo:
  - Protótipo Descartável: o objetivo é compreender os requisitos do cliente e, a partir disso, desenvolver uma melhor definição de requisitos para o sistema.
    - o protótipo se concentra em fazer experimentos com partes dos requisitos que estejam mal entendidos.



# Modelo Espiral

---

- Desenvolvido para englobar as melhores características do ciclo de vida clássico e do paradigma evolutivo.
- São avaliados riscos explicitamente e são solucionados ao longo do processo.
- Processo é representado como uma espiral em lugar de ser representado como uma sequência de atividades



# Modelo Espiral

---

- Cada loop na espiral representa uma fase do processo de software.
- Não existem fases fixas.
- Engloba as melhores características do ciclo de vida clássico como o da Prototipação, adicionando um novo elemento: a análise de riscos

# Modelo Espiral

Coleta inicial dos requisitos e planejamento do projeto

Planejamento baseado nos comentários do cliente

Avaliação do cliente

Planejamento

Análise de risco

Para cada risco do projeto identificado em P é levada a cabo uma análise detalhada.

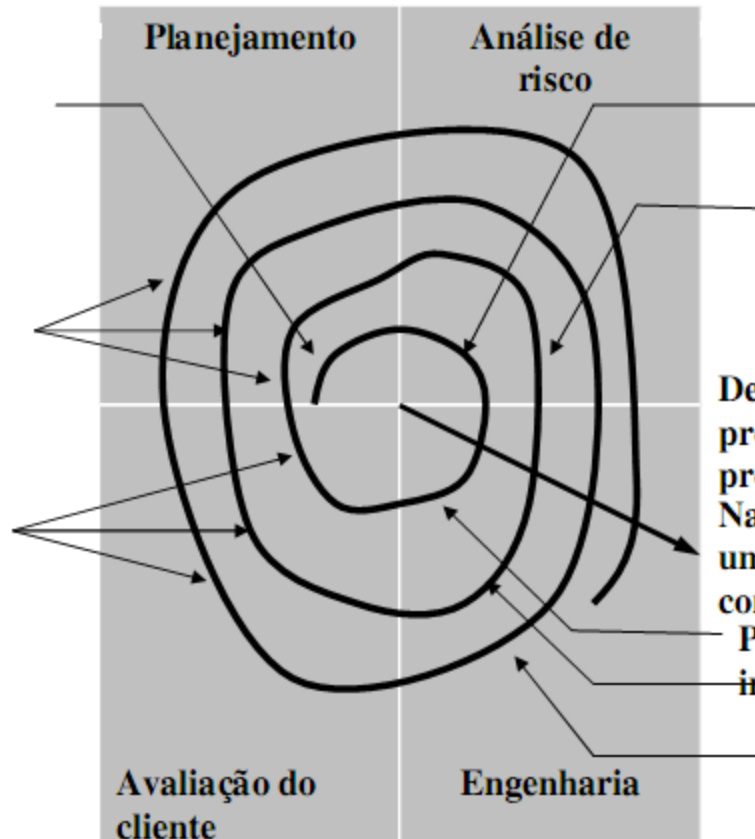
Decisão de prosseguir/não prosseguir  
Na direção de um sistema concluído

Protótipo de software inicial

Avaliação do cliente

Engenharia

Sistema construído pela engenharia





# Perguntas Comuns

---

- Qual a diferença entre processo de software e ciclo de vida?
  - Processo de software refere-se a todas as atividades, bem como relacionamentos, artefatos, ferramentas, papéis etc, necessárias para construir, entregar e manter um produto de software.
  - Já o ciclo de vida apresenta uma representação alto nível do processo de software executado (processo de software real) ou como deveria ser executado, ou seja, normalmente, ciclos de vida determinam as fases e o relacionamento entre as fases.



# Perguntas Comuns

---

- O que são recursos?
  - Recursos relacionam-se com diversos componentes do processo de software, por exemplo, técnicas, métodos e ferramentas.





# Perguntas Comuns

---

- O que são artefatos?
  - Artefato é um tipo de recurso produzido ou consumido em uma atividade. Nesse contexto, um artefato pode ser utilizado como uma entrada (matéria-prima) para uma determinada atividade e/ou como uma saída de uma atividade (resultado da execução de uma atividade).
  - Ex: código-fonte, código executável, manual de padrões, relatório de resultados, documento de requisitos, plano de trabalho, etc.



# Perguntas Comuns

---

- O que são agentes?
  - Agentes ou atores são as entidades que executam atividades por intermédio de um papel.



# Perguntas Comuns

---

- O que são papéis?
  - Papéis representam um conjunto de responsabilidades, obrigações, permissões e habilidades necessárias para executar uma atividade ou sub-atividade. Geralmente, papéis são desempenhados por agentes humanos. Um sinônimo de papel seria cargo ou função.
  - Uma atividade pode exigir diversos papéis para ser executada e um papel pode ser aplicado em diversas atividades.