



Engenharia de Software

Aula 2

Profa. Maristela Weinfurter Teixeira

Organização da Aula

▪ Processos de software

- Clássico
- Incremental
- Evolucionário
- Especializado
- RUP
- Ágil

Processos de Software

▪ Conceitos iniciais

Modelos de Processo

- Construir software é um processo de aprendizado social iterativo, e o resultado é o capital de software

- No capital de software temos a incorporação do conhecimento coletado, filtrado e organizado

Metodologia do processo

Atividades de apoio

Atividade metodológica 1		Atividade metodológica 2	
Ação 1	Ação 2	Ação 3	Ação 4
Tarefas Ação 1	Tarefas Ação 2	Tarefas Ação 3	Tarefas Ação 4



- Arcabouço para tarefas necessárias à construção de software com bom padrão de qualidade

- Um processo define a abordagem adotada para elaboração do software

- O que ganhamos ao adotar um modelo de processo?

- Estabilidade
- Controle
- Organização
- Agilidade
- Documentação
- Aderência à equipe

Modelos de Processo

- Fluxo de processo linear

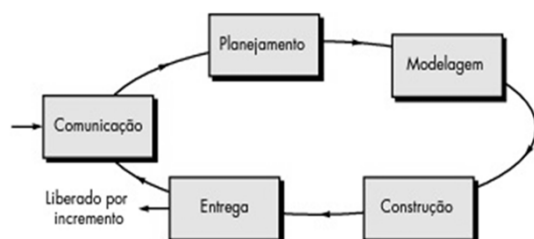


- Fluxo de processo iterativo

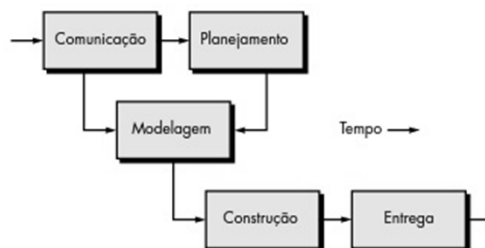




▪ Fluxo de processo evolucionário



▪ Fluxo de processo paralelo



Modelos de Processo

▪ Clássico

Clássico

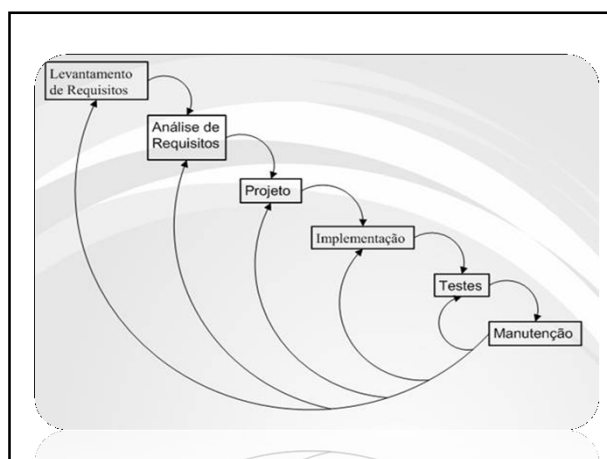
- Clássico, ou modelo cascata, é uma abordagem sequencial e sistemática

- Paradigma mais antigo da engenharia de software

- Sua eficácia é questionável
- Os projetos raramente seguem um fluxo sequencial



- Dificuldade de coleta de requisitos completa com o cliente
- Executável no final do processo



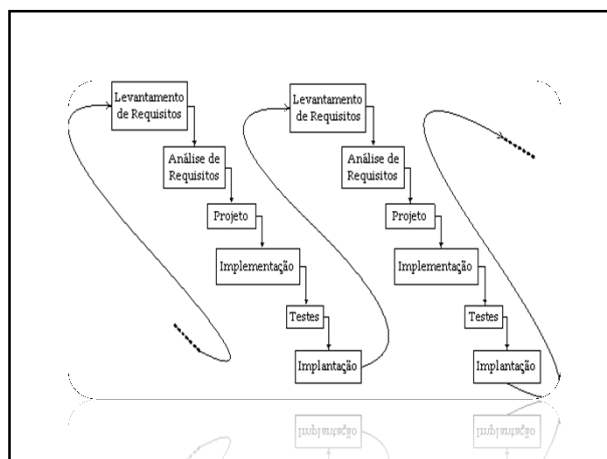
Modelos de Processo

▪ Incremental

Incremental

- A cada coleta de requisitos refina-se o projeto
- Expansão da versão anterior

- Combina fluxos de processos lineares e paralelos
- Sequencial, de forma escalonada, de acordo com o avanço do tempo





- Útil quando não há pessoal disponível para implementação completa

- Os primeiros incrementos podem ser feitos com poucos profissionais

- Os primeiros incrementos podem ser feitos para um hardware com menor capacidade

Modelos de Processo

- **Evolucionário**

Evolucionário

- Sistemas complexos que evoluem ao longo do tempo

- Necessidades de negócio e de produto mudam frequentemente



- Prazos curtos
- Mercados específicos

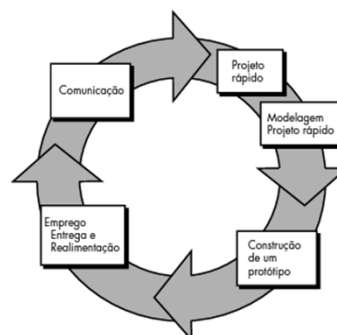
- Dois tipos:
 - prototipação
 - espiral

- ✓ Prototipação
 - Desenvolvedor encontra-se inseguro quanto à adaptabilidade do software

- Paradigma utilizado em IHC
- Requisitos obscuros

- ✓ Protótipo serve como o primeiro sistema
- ✓ Protótipo pode ser descartável

▪ Evolucionário – prototipação





✓ Espiral

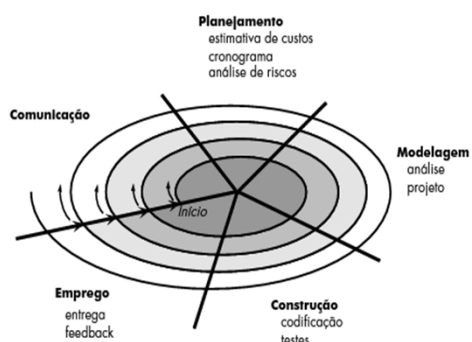
- Natureza iterativa da prototipação
- Aspectos sistemáticos do modelo clássico

- Gerador de modelos de processos dirigidos a riscos

- Abordagem cíclica voltada para ampliar, incrementalmente, o grau de definição e implementação

- Diminuição do grau de risco do projeto
- Pontos-âncoras de controle

▪ Evolucionário – espiral



✓ Espiral

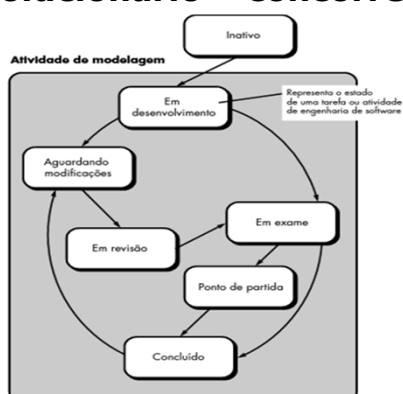
- Evolui após passar por uma série de iterações em torno da espiral



- Para software em larga escala
- Redução de riscos
- Mantém etapas, como no clássico

- Engenharia concorrente
- Representa elementos concorrentes e iterativos de qualquer modelo de processo

▪ Evolucionário – concorrentes



Modelos de Processo

- Especializado

Especializado

- Baseado em componentes
- Métodos formais
- Orientado a aspectos

- Baseado em componentes
- ✓ Incorpora características do modelo espiral



✓ Abordagem iterativa

✓ Baseada em componentes disponíveis

✓ Arquitetura acomoda componentes

✓ Reuso

• Métodos formais

✓ Especificação matemática formal

✓ Notação matemática rigorosa

✓ Correção facilitada de ambiguidade, incompletude e inconsistência

✓ Consome muito tempo e dinheiro

✓ Poucos desenvolvedores

✓ Treinamento extensivo



- Orientado a aspectos

- ✓ Paradigma novo que oferece abordagem metodológica de processos para definir, especificar, projetar e construir aspectos

- ✓ Mecanismos além de sub-rotinas e herança para localizar a expressão de uma restrição cruzada

- ✓ Não é maduro ainda
- ✓ Natureza paralela de desenvolvimento concorrente

Modelos de Processo

- RUP

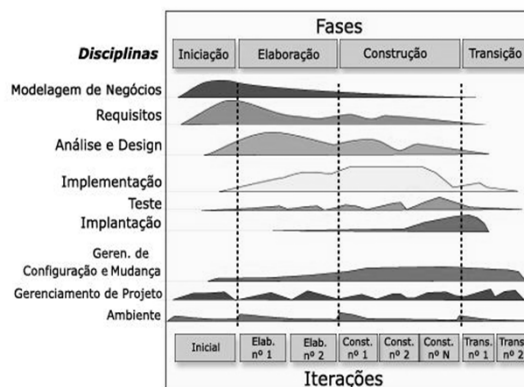
RUP

- Origem no **Processo Unificado**, de Ivar Jacobson, Grady Booch e James Rumbaugh

- Dirigido a casos de uso
- Centrado na arquitetura
- Iterativo
- Incremental



- Melhores recursos dos modelos tradicionais
- Princípios do desenvolvimento ágil
- Comunicação com o cliente
- Métodos racionalizados



Modelos de Processo

▪ Método ágil

Método Ágil

- Agilidade
- Responde apropriadamente a mudanças
 - Mudanças de:
 - ✓ equipes
 - ✓ tecnologias

- Colaborativo
- Custo de mudanças
- Achata o custo da curva de mudança

- ✓ Como criar um processo capaz de administrar a **imprevisibilidade?**
 - Adaptável



- Princípios:

- satisfazer o cliente
- pedidos de alteração
- entrega rápida
- comercial e desenvolvimento trabalham juntos

- time motivado
- comunicação efetiva
- software funcionando
- desenvolvimento sustentável

- excelência técnica
- simplicidade
- melhores arquiteturas, requisitos
- feedback e avaliação

- XP – Extreme Programming
- DAS – Desenvolvimento Adaptativo de Software
- DSDM – Método de Desenvolvimento Dinâmico de Sistemas

- Scrum
- Crystal
- FDD – Desenvolvimento Guiado por Características
- LSD – Enxuto
- AM – Modelo Ágil
- AUP – Unificado Ágil

Referências de Apoio

- SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson, 2011.
- PRESMAN, Roger. **Engenharia de Software**. 7. ed. Porto Alegre: Bookman, 2011.