

Processos de Software

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>
dcc603@gmail.com

12 Março 2012

Perguntas da Aula 2

- ★ **Inconsistente:** perguntei laranjas, respondeu pepinos
- ★★ **Incompleta:** faz algum sentido, mas não responde a pergunta
- ★★★ **Razoável:** resposta imprecisa
- ★★★★ **Boa:** falta algum detalhe relevante
- ★★★★★ **Muito Boa:** resposta interessante ou tecnicamente adequada

Respostas da Pergunta 1

- Por que leva tanto tempo para concluir um software?
 - Complexidade do software
 - Comunicação entre as partes
 - Habilidade humana
 - Processo rigoroso
 - Problemas inesperados
 - Adaptação constante

Respostas da Pergunta 2

- Por que os custos são tão altos?
 - Gastos com pessoas qualificadas
 - Testes / identificação de defeitos
 - Complexidade do software
 - Manutenção / adaptação

Respostas da Pergunta 3

- Por que não achamos todos os erros antes de entregar o software ao cliente?
 - Difícil demonstrar a ausência de erros
 - Testes incompletos
 - Falha humana
 - Difícil antecipar todas as funcionalidades
 - Diferenças entre ambiente de desenvolvimento e ambiente de produção
 - Restrição de tempo

Respostas da Pergunta 4

- Por que gasta-se tantos recursos para manter sistemas existentes (antigos)?
 - Investimento inicial grande
 - Cliente já se acostumou com o sistema atual
 - Evolução / adaptação é inevitável

Tópicos da Aula

- Processos de software
 - Definições principais
 - Atividades comuns
- Modelos de processo
 - Alguns exemplos
- Revisão

Processos de Software: Definições e Atividades

O que é um processo de software?

- Conjunto de atividades que leva ao desenvolvimento do produto software
- Um processo define
 - Quem faz, o que faz e quando fazer
 - Nem sempre diz como fazer
- Não existe um processo ideal
 - Organizações desenvolvem seus próprios processos

Atividades Comuns em Processos

1. Especificação de requisitos
2. Projeto e implementação
3. Verificação e validação
4. Evolução de software

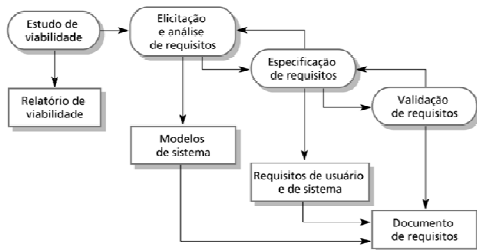
Atividades Comuns em Processos

1. **Especificação de requisitos**
2. Projeto e implementação
3. Verificação e validação
4. Evolução de software

Especificação de Requisitos

- Inclui quatro fases principais
 - Estudo de viabilidade
 - Elicitação (ou análise) de requisitos
 - Especificação de requisitos
 - Validação dos requisitos

Representação



Atividades Comuns em Processos

1. Especificação de requisitos
2. **Projeto e implementação**
3. Verificação e validação
4. Evolução de software

Projeto e Implementação

- Inclui pelo menos três fases principais
 - Projeto arquitetural (mais abstrato)
 - Projeto detalhado
 - Implementação

Desenvolvimento de Software

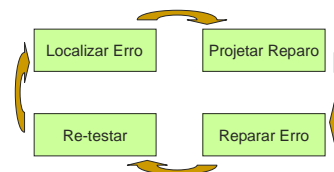
- **Projeto Arquitetural** define a estrutura modular do software, as interfaces e as estruturas de dados utilizadas
- **Projeto Detalhado** define a solução para cada módulo do projeto preliminar
- **Implementação** transcreve as decisões de projeto para uma linguagem de programação

Atividades Comuns em Processos

1. Especificação de requisitos
2. Projeto e implementação
3. **Verificação e validação**
4. Evolução de software

Verificação do software

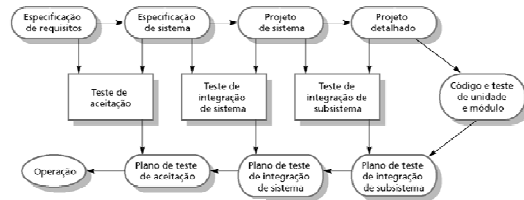
- Fase de depuração e testes



Tipos de Testes

- Teste de Componente (unitário)
 - Garantir que um componente funciona
- Teste de Sistema (integração)
 - Garantir que dois ou mais componentes funcionam juntos
- Teste de Aceitação (validação)
 - Garantir que o sistema faz o que o cliente deseja

Modelo V



Atividades Comuns em Processos

1. Especificação de requisitos
2. Projeto e implementação
3. Verificação e validação
4. **Evolução de software**

Evolução ou Manutenção

- O custo de manutenção é geralmente muito maior que o custo de desenvolvimento
- Cada vez menos sistemas são desenvolvidos "do zero"
 - Sistemas são desenvolvidos/adaptados a partir de outros sistemas
- Faz mais sentido considerar desenvolvimento e manutenção como atividades contínuas

Modelos de Processo

O que é modelo de processo?

- Descrição simplificada do processo
- Definem
 - As atividades para o desenvolvimento do software
 - Especificam os produtos de cada atividade
 - Indicam os papéis das pessoas envolvidas

Vantagens

- Oferecem um roteiro útil para o trabalho de engenharia de software
 - Mas, nenhum modelo de processo é perfeito
- Outras vantagens
 - Padronização dos artefatos
 - Melhor comunicação da equipe
 - Menos treinamento de pessoal

Exemplos de Modelos de Processo

- Modelos mais gerais
 - Modelo Cascata
 - Desenvolvimento Incremental
 - Eng. de Software orientada a Reuso
- Os modelos mais gerais não são mutuamente exclusivos
 - Organizações tendem a combinar partes de diferentes modelos

Modelos de Processo Gerais

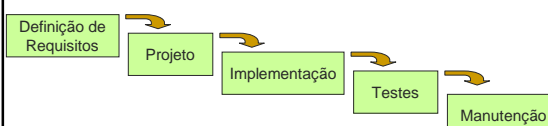
Modelo Cascata
Desenvolvimento Incremental
Baseado em Reuso

Modelos de Processo Gerais

- **Modelo Cascata**
- Desenvolvimento Incremental
- Baseado em Reuso

Modelo Cascata

- Atividades sequenciais
- Uma fase deve ser terminada para a outra começar
 - Raramente ocorre na prática



Vantagens do Modelo Cascata

- Documentação rígida (idealmente completa) em cada atividade
- Reflete abordagens adotadas em outras engenharias
- Aderência a outros modelos de processo
 - Pode ser combinado a outros modelos

Desvantagens do Modelo Cascata

- Projetos reais raramente seguem um fluxo sequencial
- Em geral, é difícil para o cliente estabelecer todos os requisitos à priori
- Difícil se adequar a mudanças inevitáveis de requisitos
- Uma versão executável somente ficará pronta na fase final do projeto

Quando Aplicar o Modelo Cascata?

- Sistemas críticos
- Quando os requisitos são bem compreendidos
- Quando há pouca probabilidade dos requisitos mudarem

Modelos de Processo Gerais

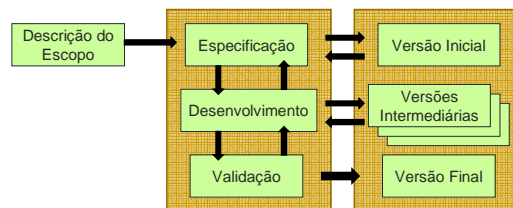
■ Modelo Cascata

■ **Desenvolvimento Incremental**

■ Baseado em Reuso

Desenvolvimento Incremental

- Atividades são intercaladas
- Objetivo: dar feedback rápido ao cliente



Vantagens

- Permite trabalhar com o cliente o entendimento dos requisitos
- Pode-se começar o sistema pelas partes melhor entendidas
- Feedback rápido ao cliente

Desvantagens

- O processo pode não ser muito claro
- A gerência do software é complicada
 - O sistema não é completamente especificado à priori
- O produto final é frequentemente mal estruturado
 - A mudança contínua tende a corromper a modularidade do sistema

Desvantagens

- O processo não é muito claro
- A gerência do software é complicada
- Os problemas do desenvolvimento incremental se tornam mais graves em sistemas críticos.
- A mudança contínua tende a corromper a modularidade

Modelos de Processo Gerais

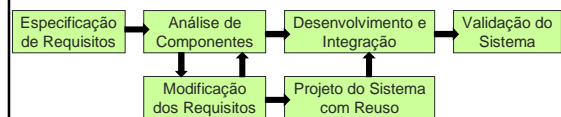
- Modelo Cascata
- Desenvolvimento Incremental
- Baseado em Reuso

Processo Orientado ao Reuso

- Baseia-se na existência de um número significativo de componentes reusáveis
- O processo se concentra na integração dos componentes reusáveis
- Inspirado na analogia com componentes de hardware
 - Exemplo: componentes elétricos / eletrônicos

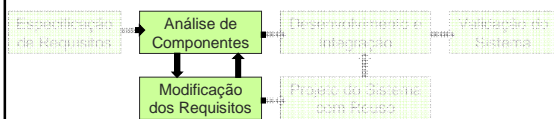
Representação

- Baseia-se na existência de um número significativo de componentes reusáveis
- O processo se concentra na integração dos componentes



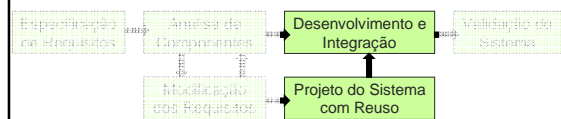
Alinhar componentes aos requisitos

- Análise de Componentes
 - Dada uma especificação, encontrar componentes que a atendam
- Modificação dos Requisitos
 - Se possível, os requisitos são adaptados aos componentes existentes



Integração dos Componentes

- Projeto do Sistema com Reuso
 - Se necessário, projeta-se novos componentes reusáveis
- Desenvolvimento e Integração
 - Desenvolvimento de novos componentes
 - Integração de **todos** os componentes



Vantagens

- Reduz a quantidade de software a ser desenvolvido
- Espera-se reduzir os custos e os riscos
- Espera-se uma entrega do produto mais rápida ao cliente

Desvantagens

- Pode-se desenvolver um produto que não atenda aos requisitos do cliente
- Pode ser mais difícil evoluir os sistemas
 - Componentes de terceiros
- A gerência de versões dos componentes pode ser complexa

Qual modelo de processo usar?

- Sistemas Críticos
 - Sugerido um modelo de processo mais estruturado / rigoroso como o Modelo Cascata
- Sistemas de Negócios (requisitos mudam com frequência)
 - Sugerido um modelo de processo ágil e flexível como o Desenvolvimento Incremental ou o Baseado em Reuso

Resumo

- Definições básicas
 - o que é processo de software?
 - o que são métodos de processo?
- Algumas atividades comuns a diferentes processos
 - Especificação, projeto e implementação, validação, e evolução

Bibliografia Principal

- Ian Sommerville. Engenharia de Software, 9ª Edição. Pearson Education, 2011.
 - Seção 2.1 Modelos de Processo
 - Seção 2.2 Atividades de Processo
 - Seção 2.3 Lidando com Mudanças
- Ian Sommerville. Engenharia de Software, 8ª Edição. Pearson Education, 2007.
 - Seção 4.1 Modelos de Processo
 - Seção 4.2 Iteração de Processo
 - Seção 4.3 Atividades de Processo