

Revisão para Prova 1

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

dcc603@gmail.com

22 Abril 2013

Aula 1: Apresentação

- Livro
- Datas
- Monitoria
- Avaliação
- Agenda do Curso

Aula 2: Motivação

- Motivação e Conceitos Fundamentais
- Visão da Engenharia de Software em Camadas
- Atividades do Desenvolvimento de Software
- Evolução de Software (Leis de Lehman)

O que é software?

- Programa de computador + Documentação
- Classificação fundamental
 - Produtos genéricos (ex. MS Office)
 - Produtos encomendados (ex. Software de Controle da Locadora do Zé)

O que é Engenharia de Software?

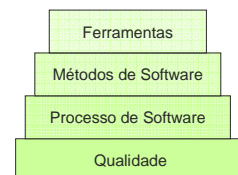
A Engenharia de Software é uma disciplina de engenharia relacionada a todos os aspectos de produção de software.

Ian Sommerville

- Foco no desenvolvimento de software de **alta qualidade** dentro de **custos adequados**.
 - Atender necessidades do cliente

Eng. de Software em Camadas

- A engenharia de software inclui
 - Processo
 - Métodos
 - Ferramentas



- Pode ser organizada em camadas

[Atividades de Desenvolvimento]

1. Especificação de requisitos
2. Projeto de Software
3. Implementação
4. Validação do software
5. Evolução de software

[Bibliografia da Aula 2]

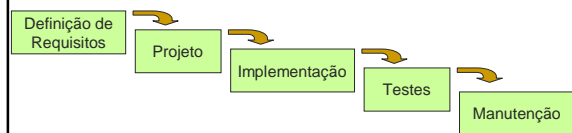
- Roger Pressman. **Engenharia de Software: Uma Abordagem Profissional**, 7ª Edição. McGraw-Hill, 2011.
 - Capítulo 1
- Ian Sommerville. **Engenharia de Software**, 9ª Edição. Pearson Education, 2011.
 - Capítulo 1

[Aula 3: Processos]

- Atividades comuns em processos de software
- Modelos de processo
- Modelos de processo que lidam com mudanças

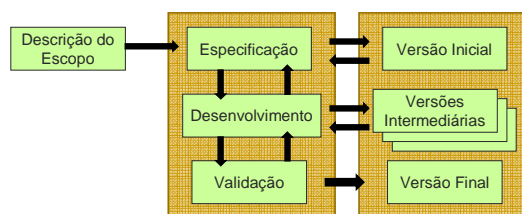
[Modelo Cascata]

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



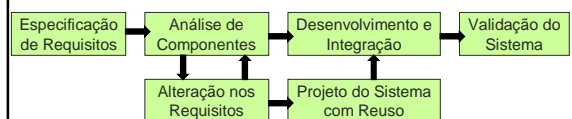
[Desenvolvimento Incremental]

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



[Baseado em Reuso]

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

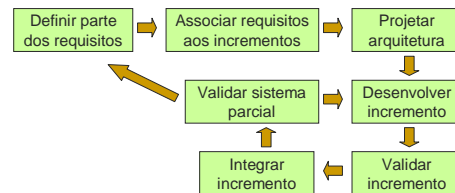


[Prototipação]

- É geralmente usada junto com outro modelo de processo
- Planeja e modela rapidamente um protótipo
 - Mais comum na definição de interfaces com os usuários (telas)
- Começa com os requisitos menos compreendidos
 - Objetivo: entender os requisitos

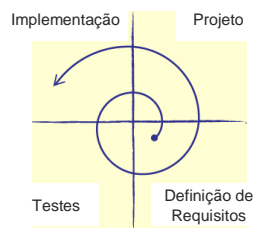
[Entrega Incremental]

- Combina elementos do modelo cascata aplicados de maneira iterativa



[Modelo Espiral]

- Combina elementos dos modelos Incremental e Prototipagem
 - É sequência adotada do Modelo Cascata
- Software é desenvolvido em versões
 - Prototipagem nas primeiras versões
 - Incremental nas últimas versões



[Bibliografia da Aula 3]

- Ian Sommerville. **Engenharia de Software**, 9ª Edição. Pearson Education, 2011.
 - Cap. 2

[Aula 4: Métodos Ágeis]

- Métodos ágeis de desenvolvimento de software
- Programação Extrema (XP)
- Scrum

[Manifesto Ágil]

- Passamos a valorizar:
 - **Indivíduos e interações** mais que processos e ferramentas
 - **Software em funcionamento** mais que documentação abrangente
 - **Colaboração com o cliente** mais que negociação de contratos
 - **Responder a mudanças** mais que seguir um plano

<http://agilemanifesto.org/>

Práticas de XP

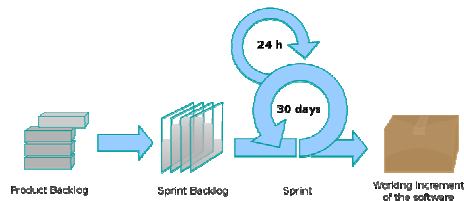
- Planejamento incremental
- Pequenas releases
- Projeto simples
- Desenvolvimento dirigido por testes
- Refatoração
 - Reestruturação constante de código

Práticas de XP

- Programação por pares
- Propriedade coletiva do código
 - Qualquer um pode modificar qualquer coisa (não há ilha de conhecimento)
- Integração contínua
- Ritmo sustentável
- Cliente sempre disponível

Ciclo do Processo

- Planejamento (*pre-game phase*)
- Desenvolvimento (*game phase*)
- Encerramento (*post-game phase*)



Bibliografia da Aula 04

- Ian Sommerville. **Engenharia de Software**, 9ª Edição. Pearson Education, 2011.
 - Cap. 3
- A. Koscianski e M. Soares. **Qualidade de Software**, 2ª Edição. Novatec, 2006.
 - Seção 10.3 Metodologias Ágeis

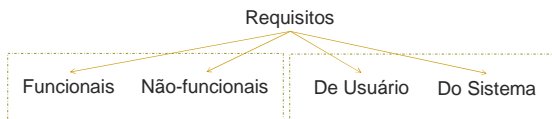
Aula 5: Exercício iTeamwork

- Criar um projeto de software
 - Tema: Sistema Bancário
- Entregar perguntas por email
 - dcc603@gmail.com
 - Título da mensagem: [Exercicio 1] XXX e YYY
 - XXX e YYY são os seus nomes

Aula 6: Dúvidas Vídeo-Aulas

- Juliana ficou na sala de aula
 - Tirar dúvidas sobre criação e edição de vídeo aulas

[Aula 7: Eng. de Requisitos]



[Requisitos de Usuário]

- Descreve as funções e restrições do sistema de forma abstrata
 - Inteligível pelo usuário / cliente
- Ponto de vista das necessidades da empresa
 - Não indica uma solução
- Escrito em linguagem natural com diagramas simples (ex. tabelas)

[Requisitos de Sistema]

- São descrições mais detalhadas que os requisitos do usuário
- Devem ser padronizados, completos e consistente
 - Usados pela equipe de desenvolvimento
 - Fazem parte do contrato

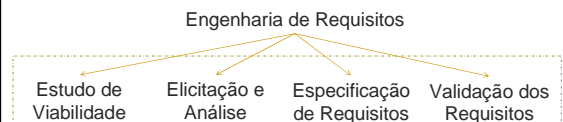
[Requisitos Funcionais]

- Descrevem explicitamente as funcionalidades e serviços do sistema
- Documenta
 - como o sistema deve reagir a entradas específicas
 - como deve se comportar em determinadas situações
 - o que o sistema não deve fazer

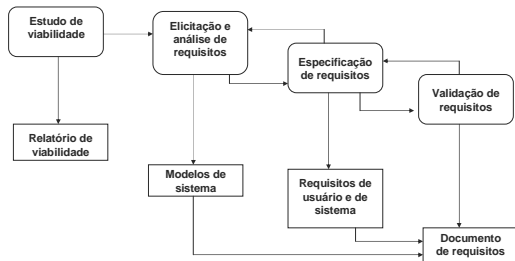
[Requisitos Não-Funcionais (RNF)]

- Definem propriedades e restrições do sistema
 - Exemplos: segurança, desempenho, espaço em disco
- Podem ser do sistema todo ou de partes do sistema
- Requisitos não-funcionais podem ser mais críticos que requisitos funcionais
 - Se não satisfaz, o sistema é inútil

[Aula 8: Eng. de Requisitos]



[Eng. de Requisitos: Processo]



[Bibliografia das Aulas 7 e 8]

- Ian Sommerville. **Engenharia de Software**, 9ª Edição. Pearson Education, 2011.
 - Capítulo 4

[Aula 9: UML e Casos de Uso]

- Introdução a UML
- Visão geral de alguns diagramas
- Diagrama de Casos de Uso

[Alguns Diagramas UML]

- Diagramas Estruturais (Estáticos)
 - Diagrama de Casos de Uso
 - Diagrama de Classes
 - Diagramas de Objetos
 - Diagrama de Componentes, etc.
- Diagramas Dinâmicos
 - Diagrama de Sequência
 - Diagrama de Estados
 - Diagrama de Atividades
 - Diagrama de Colaboração, etc.

[Diagrama de Casos de Uso]

- Atores
 - Quem executa a funcionalidade
- Casos de Uso
 - Qual é a funcionalidade
- Relacionamentos
 - Como atores e casos de uso se relacionam

[Bibliografia da Aula 9]

- G. Booch, J. Rumbaugh, I. Jacobson. **UML, Guia do Usuário**. 2ª Ed., Editora Campus, 2005.
 - Capítulos 1, 17 e 18
- M. Fowler. **UML Essencial**, 2ª Edição. Bookmann, 2000.
 - Capítulo 3

Aula 10: Exercício

1. Criar (um ou mais) diagramas de casos de uso
 - Usar o ArgoUML para criar os diagramas
 - C:\Program Files\ArgoUML -> ArgoUML
2. Documentar cada Caso de Uso utilizando a técnica de cenários
 - Usar o Word (ou outro programa) para descrever os cenários

Aula 11: Explicações

- Faltou luz, aula acabou mais cedo
 - Explicação dos exercícios e dos critérios de participação online

Aula 12: Arquitetura de Software

- Arquitetura de Software
- Padrões arquiteturais
 - Arquitetura em Camadas
 - Dutos e Filtros
 - Repositório Compartilhado
 - Cliente-Servidor
 - Modelo-Visão-Controle
 - Microkernel

Projeto Arquitetural

- O projeto é uma atividade criativa
 - Cada arquiteto tem sua própria maneira de projetar o software
- Projeto arquitetural faz a ligação entre
 - Requisitos (domínio do problema) e
 - Projeto detalhado (domínio da solução)

Padrões Arquiteturais

- Da desordem a estrutura
 - Layered Architecture (Arquitetura em Camadas)
 - Blackboard (Arquitetura de Repositório)
 - Pipes and Filters (Dutos e Filtros)
- Sistemas distribuídos
 - Client-Server (Cliente-Servidor)
 - Broker
- Sistemas interativos
 - Model-View-Controller (MVC)
 - Presentation-Abstraction-Control
- Sistemas adaptáveis
 - Microkernel
 - Reflection

Discutidos no livro do Sommerville

Bibliografia da Aula 12

- Ian Sommerville. **Engenharia de Software**, 9a. Edição. 2011.
 - Cap. 6
- F. Buschmann et al. **Pattern-Oriented Software Architecture: A System of Patterns**. John Wiley & Sons, 1996.
 - Cap. 2 Architectural Patterns

[Aula 13: Diagrama de Classes]

- Projeto orientado a objetos
 - Dos requisitos ao projeto
- Diagrama de Classes
 - Classes e interfaces, métodos e atributos, visibilidade, multiplicidade e notas
- Relacionamentos
 - Herança, associação, agregação, composição e dependência

[Projeto Orientado a Objetos]

- Maneira natural de visualizar o software
 - Documentação de alto nível
 - Comunicação entre a equipe
- Modela o software semelhante ao mundo real - usando objetos
- **Objetos** são modelados em termos de seus **atributos** e comportamento (**métodos**)

[Diagrama de Classes]

- Serve de apoio para a maioria dos outros diagramas
- Define a estrutura das classes do sistema
- Estabelece como as classes se relacionam

[Bibliografia da Aula 13]

- Ian Sommerville. **Engenharia de Software**, 9a. Edição. 2011.
 - Cap. 7
- G. Booch, J. Rumbaugh, I. Jacobson. **UML, Guia do Usuário**. 2ª Ed., Editora Campus, 2005.
 - Capítulos 4, 8 e 9
- M. Fowler. **UML Essencial**, 2a Edição. Bookmann, 2000.
 - Capítulos 4 e 6

[Aula 14: Diagramas Dinâmicos]

- Diagrama de Sequência
- Diagrama de Colaboração
- Diagrama de Atividades

[Diagrama de Sequência]

- Preocupa-se com a ordem temporal em que as mensagens são trocadas
- Pode ser usado para detalhar um Caso de Uso
- Identifica
 - O evento gerador da funcionalidade modelada (ator responsável pelo evento)
 - Os objetos envolvidos na ação

Ver Informação de Paciente

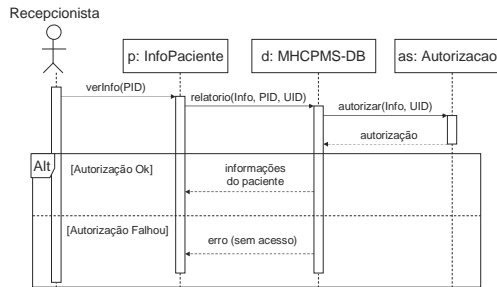


Diagrama de Colaboração

- Não se preocupa com a temporalidade
- Define a estrutura de como os objetos estão vinculados
 - Como o Diagrama de Classes
- Indica quais mensagens são trocadas entre objetos
 - Como o Diagrama de Sequência

Diagrama de Colaboração

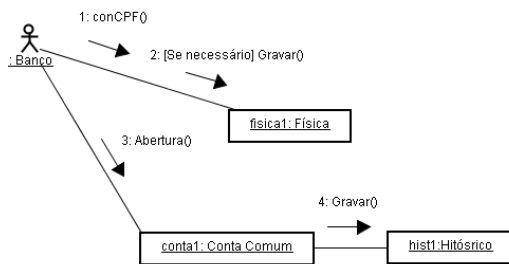
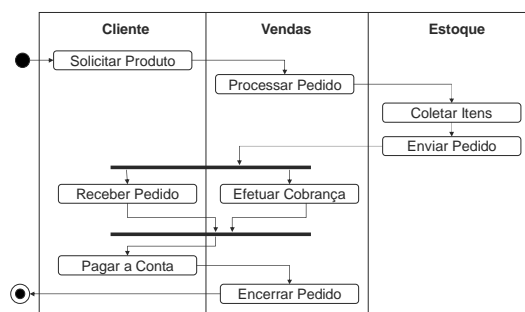


Diagrama de Atividades

- Mostram as atividades que compõem um processo do sistema e o fluxo de controle
 - Decompõe um processo em suas atividades
- Modelam a execução de atividades sequenciais ou concorrentes

Diagrama de Atividades



Bibliografia da Aula 14

- Ian Sommerville. **Engenharia de Software**, 9a. Edição. 2011.
 - Cap 5.
- G. Booch, J. Rumbaugh, I. Jacobson. **UML, Guia do Usuário**. 2ª Ed., Editora Campus, 2005.
 - Capítulos 16, 19 e 20
- M. Fowler. **UML Essencial**, 2a Edição. Bookmann, 2000.
 - Capítulo 5 e 9

[Datas Importantes]

- **22/04/2013 (HOJE!) até 23:59 hrs**
 - Informar por email grupo e tema escolhido para o Trabalho Prático (TP)
 - Entregar os arquivos de vídeo-aula
 - Aulas 1 a 20 de acordo com a numeração do udemy
 - Enviar por email um link para baixar ou entregar via pen-drive ou CD