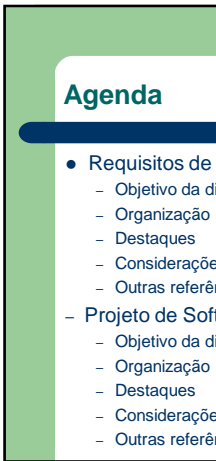




SWEBOK

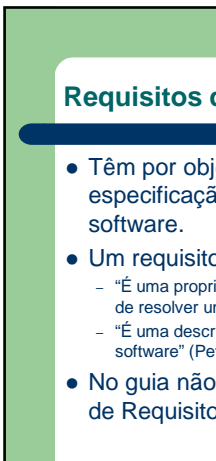
Requisitos de Software & Projeto de Software

Profº. Msc. Rodrigo Santos



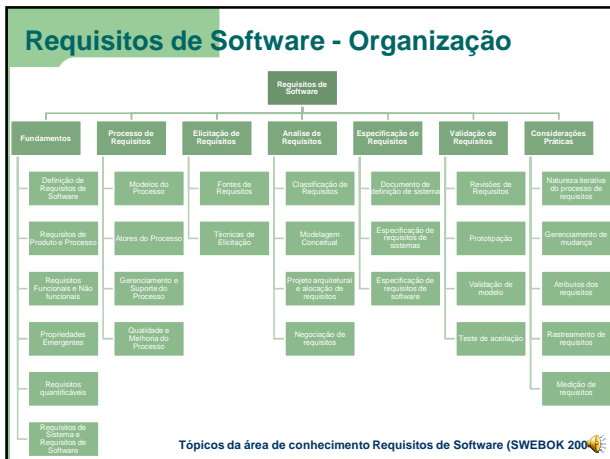
Agenda

- Requisitos de Software
 - Objetivo da disciplina
 - Organização no guia
 - Destaques
 - Considerações práticas
 - Outras referências
- Projeto de Software
 - Objetivo da disciplina
 - Organização no guia
 - Destaques
 - Considerações práticas
 - Outras referências



Requisitos de Software - Objetivo

- Têm por objetivo tratar a elicitação, análise, especificação e validação dos requisitos de software.
- Um requisito de software é:
 - “É uma propriedade que precisa ser exibida com o objetivo de resolver um problema do mundo real” (Swebok, p. 33)
 - “É uma descrição de um recurso de um produto de software” (Peters & Pedrycz, p. 102)
- No guia não é utilizado o termo “Engenharia de Requisitos”



Requisitos de Software - Destaques

- **Requisito de Sistema x Requisito de Software**
 - **Requisito de Sistema**
 - Contempla vários itens como hardware, software, firmware, pessoas, informações, etc.
 - Descrição do funcionamento do hardware, software e usuário para a solução de um problema
 - Ex: Sistema de rastreamento de veículos

www.raytel.com.br Profº. Msc. Rodrigo Santos

Requisitos de Software - Destaques

- **Requisito de Sistema x Requisito de Software**
 - **Requisito de Software**
 - Pode ser derivado do requisito de sistema
 - Trata o escopo do software e suas funcionalidades
 - Ex.: O software deverá ordenar a lista de alarmes por ordem de data.

Profº. Msc. Rodrigo Santos

Requisitos de Software - Destaques

- Especificação de Requisitos
 - Contém o que o software deve fazer e serve de “contrato” entre a equipe de desenvolvimento e o cliente.
 - É base para todas as demais atividades
 - Existe um padrão da IEEE (IEEE-830/98), que trata de como desenvolver e dos principais itens de uma especificação.

Profº. Msc. Rodrigo Santos



Requisitos de Software – Considerações Práticas

- Os requisitos não são imutáveis
 - Por isso se faz necessário documentar
- Geralmente problemas na especificação se espalham por todo o projeto, e o custo de correção aumenta conforme a fase em que se detecta o problema.
- A maior dificuldade é saber o que documentar e em qual nível

Profº. Msc. Rodrigo Santos



Requisitos de Software – Considerações Práticas

- Rastreamento é fundamental
 - Controle de mudanças
 - Apoio de ferramentas em projetos muito grandes pode ser fundamental
 - Para pequenos projetos até uma planilha Excel atenderia.
- Os requisitos devem ser verificados e podem ser medidos (Ex.: Estimativas de tamanho (FPA))

Profº. Msc. Rodrigo Santos



Requisitos de Software – Outras Referências

- Requirements Engineering: A Good Practice Guide (1997) - Ian Sommerville, Pete Sawyer
- Effective Requirements Practices (2001) - Ralph Rowland Young
- Writing Better Requirements (2002) - Ian Alexander, Richard Stevens
- Managing Software Requirements: A Use Case Approach (2003) - Dean Leffingwell, Don Widrig, Ed. Yourdon.

Profº. Msc. Rodrigo Santos



Projeto de Software - Objetivo

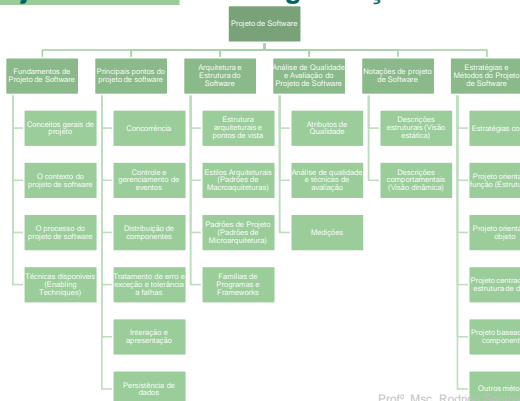
- Projeto (*Design*) no contexto de engenharia;
 - 4 Constr Representação gráfica e escrita com orçamento de uma obra que se vai realizar (in Michaelis, 2009)
- “O projeto de software deve descrever a arquitetura de software – como é, como o software é decomposto e organizado em componentes – e as interfaces entre estes componentes. Deve também descrever estes componentes em um nível de detalhe que permita sua construção.” (Swebok 2004, p. 51).

Profº. Msc. Rodrigo Santos



Projeto de Software - Organização

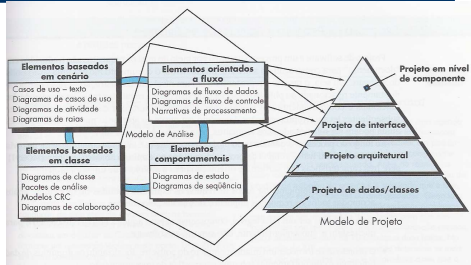
Tópicos da área de conhecimento Projetos de Software (SWEBOX 2004)



Profº. Msc. Rodrigo Santos



Projeto de Software - Destaque



(Pressman, 2006, p. 187)

Profª. Msc. Rodrigo Santos



Projeto de Software - Destaques

- Padrões de projeto - OO
 - Excelente saída para diversas questões que já foram tratadas e testadas em vários outros projetos
 - São categorizados em:
 - Padrões de Criação
 - Padrões Estruturais
 - Padrões de Comportamento

Profª. Msc. Rodrigo Santos



Projeto de Software - Destaques

- Padrões de Criação - OO:
 - Ajudam a tornar um sistema independente de como seus objetos são criados, compostos e representados.
 - A ênfase se desloca da codificação de maneira rígida de um conjunto fixo de comportamentos para a definição de um conjunto menor de comportamentos fundamentais que possam ser especializados.
 - Ex.: Abstract Factory, Builder, Factory Method, Prototype, Singleton

Profª. Msc. Rodrigo Santos



Projeto de Software - Destaques

- Padrões comportamentais - OO:
 - Se preocupam com algoritmos e a atribuição de responsabilidades entre objetos.
 - Podem ser descritos como um grupo de objetos pares que cooperam para a execução de uma tarefa que nenhum objeto sozinho poderia executar por si mesmo.
 - Ex.: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor

Profº. Msc. Rodrigo Santos



Projeto de Software - Destaques

- Padrões estruturais - OO:
 - Se preocupam com a forma como classes e objetos são compostos para formar estruturas maiores.
 - Trata diversos dos problemas de comunicação e interface entre classes e objetos.
 - Ex.: Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy

Profº. Msc. Rodrigo Santos



Projeto de Software – Considerações Práticas

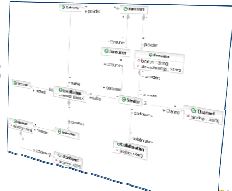
- Ainda falta maturidade ao desenvolvimento de software
 - Alguém constrói um prédio sem todos os projetos de engenharia?
 - Quantos engenheiros/projetistas de software você conhece?
 - Os custos de se refazer uma construção (ou parte de uma) por erro de projeto podem ser muito altos ou inviáveis, e no software é diferente?

Profº. Msc. Rodrigo Santos



Projeto de Software – Considerações Práticas

- Uma das grandes evoluções nos últimos anos foi o advento da UML:
 - Simplicidade;
 - Unificado;
 - Independente de linguagem;
 - Amplamente aceito pelo mercado



http://www.ibm.com/developerworks/rational/library/05419_sos

Requisitos de Software – Outras Referências

- Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos (2005) – Erick Gama, et al
- Utilizando UML e Padrões (2007) – Craig Larmn
- Uml 2.0 And The Unified Process (2005) – Jim Arlow, Ila Neustadt
- OMG - <http://www.uml.org/>

Profº. Msc. Rodrigo Santos



Referências

- Guide to the Software Engineering Body of Knowledge, 2004. Disponível em: www.swebok.org.
- Engenharia de Software, 2006, Roger S. Pressman.
- Engenharia de Software: Teoria e Prática, 2001, James F. Peters, Witold Pedrycz
- Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos, 2000, Erick Gama, et al

Profº. Msc. Rodrigo Santos

