PARTICIPANTES DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Projeto de Programas – PPR0001

Componente Humano / Participantes

- **Gerente de projeto**
  - Gerência ou coordenação do projeto
  - Define quem faz o quê e quando
  - Estipula orçamento e tempo
  - Definir processo de desenvolvimento / metodologias
  - Define e busca recursos de hardware e software
  - Realizar o acoplamento das atividades

Componente Humano / Participantes

- **Analista**
  - Deve ter conhecimento do domínio do negócio
  - Não precisa ser um especialista , mas deve ter um conhecimento básico na área de domínio para se comunicar com especialistas
  - Entender as necessidades dos clientes e repassar a equipe
  - Deve ter conhecimento relativos à modelagem de sistemas
    - Tradutor: linguagem dos especialistas do domínio e dos desenvolvedores
  - Ter bom relacionamento interpessoal (+ importante que tecnológico)
  - Ética profissional: contato com informações sigilosas
  - Analista de negócios: entender o que o cliente faz, por que faz, e como o processo pode ser otimizado por um sistema
  - Analista de sistemas: traduz necessidades do usuário em características de um produto de software

Componente Humano / Participantes

- **Projetistas**
  - Projetar alternativas de solução do problema resultante da análise, i.e. adicionam aspectos tecnológicos a tais modelos
  - Gerar especificações de uma solução computacional detalhada
  - Na prática podem existir diversos tipos:
    - Projetistas de Interface
    - Projetistas de Rede
    - Projetistas de Banco de Dados
    - (...)

Componente Humano / Participantes

- **Arquitetos de software**
  - Elaborar a arquitetura de um software como um todo
  - Define quais serão as subdivisões do sistema e como serão as interfaces entre eles
  - Deve ser capaz de tomar decisões técnicas detalhadas
    - E.g. decisão sobre um aspecto em relação ao desempenho do sistema
  - Geralmente presentes em grandes equipes e projetos complexos

Componente Humano / Participantes

- **Programadores**
  - Responsáveis pela implementação do sistema
  - Proficientes em uma ou mais LPs e capazes de ler modelos de projeto
  - Participam principalmente dos processos finais do desenvolvimento
  - !! um bom programador não necessariamente é um bom analista
  - !! um bom analista não necessariamente é um bom programador

Componente Humano / Participantes

- ***Testers***
  - Avaliam se as funcionalidades do software estão em acordo com as especificações realizadas (geralmente utilizam *checklists*)
  - Podem realizar testes de qualidade (desempenho e confiabilidade)
- **Avaliadores de Qualidade**
  - Avaliam desempenho e confiabilidade do software
  - Avaliam qualidade e adequação durante o desenvolvimento

Componente Humano / Participantes

- **Especialistas do domínio**
  - Pessoas que possuem muito conhecimento no domínio do projeto
  - São os clientes:
    - Cliente contratante: indivíduo(s) que solicitaram o desenvolvimento
    - Cliente usuário: indivíduo(s) que utilizará o sistema
  - Em projetos que são desenvolvidos para a massa, são escolhidos representantes para clientes:
    - Equipe de marketing
    - Usuários comuns do tipo de software sendo desenvolvido

FERRAMENTAS NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Projeto de Programas – PPR0001

Ferramenta para Modelagem

- Evolução de computadores -> necessidade de evolução na modelagem de sistemas e processos de desenvolvimento
  - Exemplo mais recente: construção de um software monoprocessado é igual a produção de um software com processamento paralelo?

- Histórico:
  - **1950/60**: sistemas simples; técnicas de modelagem mais simples; desenvolvimento *ad hoc* (direto ao assunto); fluxogramas e diagramas de módulos
  - **1970**: computadores avançados e acessíveis (expansão do mercado computacional); sistemas mais complexos; surgimento da programação estruturada; modelos mais robustos começavam a surgir

Ferramenta para Modelagem

- Histórico:
  - **1980**: computadores mais avançados e baratos; interface gráfica; consolidação da análise estruturada
  - **1990-1996**: surgimento do paradigma de orientação a objetos. Várias propostas de técnicas surgiram:
    - OOAD – *Object-Oriented Analysis and Design*
    - *Booch Method*
    - OOSE – *Object-Oriented Software Engineering*
    - OMT – *Object Modeling Technique*
    - *Responsibility Driven Design*
    - *Fusion*
    - Problema: diferentes notações gráficas para representar uma mesma perspectiva

Ferramenta para Modelagem

- Histórico:
  - **1996-2000**: percebeu-se a necessidade de um padrão de modelagem para indústria e academia; surge a UML.

- UML (Linguagem de Modelagem Unificada)
  - Grady Booch, James Rumbaugh e Ivar Jacobson (“três amigos”)
  - Aproveitou as melhores notações existentes na época
  - Aprovada como padrão em 1997
  - Em desenvolvimento / evolução: versão atual 2.0
  - É uma linguagem visual para modelar sistemas O.O
  - Pode representar diversas perspectivas de um sistema

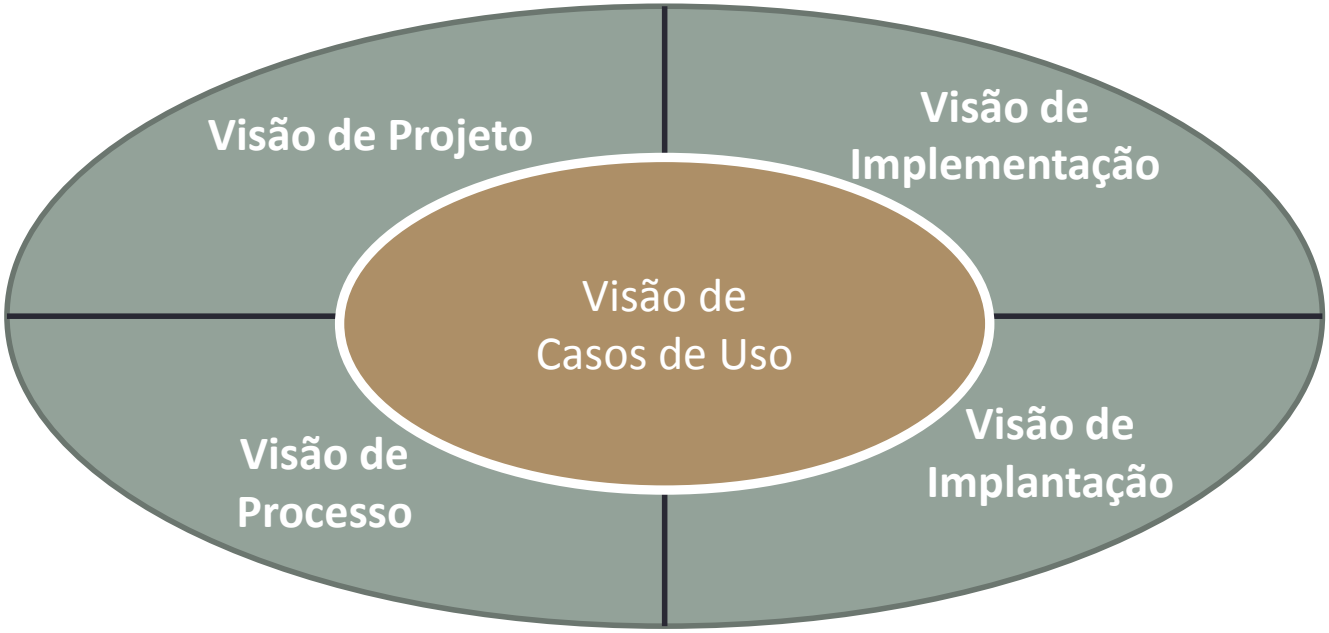
Ferramenta para Modelagem

- UML (Linguagem de Modelagem Unificada)
  - Pode ser utilizada para vários tipos de processos de desenvolvimento
    - Produz-se **artefatos**
  - Cada elemento possui uma sintaxe e uma semântica
    - Sintaxe: forma de desenho
    - Semântica: significado do elemento
  - É extensível: pode se adaptar às características específicas de cada projeto

Ferramenta para Modelagem

Visões de um sistema

- Autores da UML sugerem que um sistema pode ser visto a partir de cinco visões interdependentes:



Ferramenta para Modelagem

Visões de um sistema

- Visão de Casos de Uso: descrição do sistema do ponto de vista externo; conjunto de interações entre o sistema e os agentes externos; visão inicial que direciona o desenvolvimento das outras visões
- Visão de Projeto (design): ênfase nas características do sistema - estrutura e comportamento - e nas funcionalidades visíveis
- Visão de Processo: ênfase nas características de concorrência, sincronia e desempenho do sistema
- Visão de Implementação: gerenciamento de versões do sistema, e do agrupamento dos módulos/componentes
- Visão de Implantação: distribuição física do sistema em seus subsistemas e conexão entre as partes

Ferramenta para Modelagem

Visões de um sistema

- Nem sempre todas as visões são necessárias:
  - Um sistema a ser instalado em um ambiente monoprocessado
  - Um sistema construído a partir de um único processo

Prototipagem

- Protótipo = esboço de alguma parte do sistema
- Serve de complemento à análise
- Produtos comuns:
  - Telas do sistema
  - Módulos funcionais sem interface rebuscada
  - Esboço do sistema completo
- Para telas, uso comum de linguagens de programação visuais:
  - Delphi, PowerBuilder, Visual Basic, Front Page, ...
- Para módulos, uso comum de linguagens script:
  - Ex.: Python

Prototipagem

- Protótipo são utilizados na validação de requisitos:
  - Protótipos são mais concretos do que modelos
  - Críticas ou sugestões ao protótipo são corrigidas/incorporadas
- Ciclo de prototipagem continua até ser aceito pelos clientes
  - Após aceito, o protótipo pode servir de base para o software final ou ser descartado
- A técnica de protótipos é opcional mas é muito utilizada, principalmente quando os requisitos são de difícil entendimento

Ferramentas no processo de desenvolvimento

- O processo de desenvolvimento de software é complicado e altamente cooperativo
- O uso de ferramentas auxiliares podem ajudar:
  - Na construção de modelos do sistema
  - Na integração do trabalho da equipe
  - No gerenciamento do andamento de desenvolvimento
  - (...)

- Softwares para suporte ao ciclo de desenvolvimento:
  - Ferramentas CASE
  - Ambientes de Desenvolvimento – IDE (*Integrated Development Environment*)

Ferramentas no processo de desenvolvimento

Ferramentas CASE:

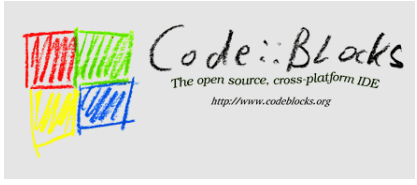
- *Computer Aided Software Engineering* – Engenharia de software auxiliada por computador
  - Criação de diagramas (XMI – XML Metadata Interchange) [UML]
  - Engenharia *Round-Trip*: interage com o código-fonte do sistema
    - Engenharia Direta: diagramas → código
    - Engenharia Reversa: código → diagramas
  - Rastreamento de requisitos: facilitar a localização de artefatos criados com base em um dado requisito.
  - Ferramenta: JUDE/ASTAH

Ferramentas no processo de desenvolvimento

Ambientes de Desenvolvimento:

- Possibilitam codificação com diversos funcionalidades adicionais que facilitam este processo:

  - Depuração de código-fonte (facilita procura por erros de lógica)
  - Pré-Compilação (procura-se por erros no momento da escrita)
  - Refatoração: modificações no código que não alterem seu comportamento (renomear variáveis, classes, ...)
  - Ferramenta: Dev-C++, CodeBlocks, NetBeans, Eclipse



Ferramentas no processo de desenvolvimento

Outras Ferramentas:

- Realização de testes automatizados
- Gerenciamento de versões de documentos (dropbox, SVN)
- Monitoração e averiguação de desempenho (tempo de execução, uso de memória, tráfego de dados
- Tarefas de gerenciamento (jxProject, OpenProj, Project Planner)

Bibliografia

- **Básica:**

  BEZERRA, E. Princípios de Análise e Projetos de Sistemas com UML. Rio de Janeiro: Campus, 2003.

  PRESSMAN, R.S. Engenharia de Software. São Paulo: Makron Books, 2002.

  SOMMERVILLE, I. Engenharia de Software. São Paulo: Addison Wesley, 2003.

- **Complementar:**

  WARNIER, J. Lógica de Construção de Programas. Rio de Janeiro: Campus, 1984.

  JACKSON, M. Princípios de Projeto de Programas. Rio de Janeiro: Campus, 1988.

  PAGE-JONES, M. Projeto Estruturado de Sistemas. São Paulo: McGraw-Hill, 1988.