

Engenharia de Software: conceitos e aplicações

Prof. Tiago Eugenio de Melo, MSc
tiagodemelo@gmail.com

Objetivos da aula

- Apresentar os conceitos de Engenharia de Software e explicar a sua importância.
- Explicar as questões chaves sobre Engenharia de Software.
- Apresentar os principais processos de desenvolvimento de software.

Sumário

- Introdução à Engenharia de Software
- Princípios de Engenharia de Software
- Processos de Engenharia de Software
- Questões de revisão
- Referências bibliográficas

Motivação

- A economia de todos países dependem do uso de software.
- Cada vez mais, o controle dos processos tem sido feito por software.
- Atualmente, os custos de software superam os custos de hardware.
- A manutenção de software é onde se tem os maiores gastos. Principalmente em sistemas de vida longa.

Conceitos

- O que é software?
 - Programas de computador e documentação associadas.
 - O software pode ser classificado em:
 - Sistemas genéricos, produzidos e vendidos no mercado a qualquer pessoa que queira comprá-los.
 - Sistemas específicos, encomendados especificamente por um determinado cliente.

Conceitos

- Tipos de sistema de software:
 - Sistemas legados.
 - Sistemas de tempo real.
 - Sistemas embarcados (embutidos).
 - Sistemas científicos.
 - Sistemas para WEB.
- Ciclo de vida do produto de software
 - Começa na concepção do problema (solicitação do usuário).
 - Termina quando o sistema sai de uso.

Conceitos

- O que é Engenharia de Software?
 - É uma das áreas da Engenharia que trata dos aspectos de produção de software.
 - O seu objetivo é estabelecer uma sistemática abordagem de desenvolvimento, através de ferramentas e técnicas apropriadas, dependendo do problema a ser abordado, considerando restrições e recursos disponíveis.

Conceitos

- Quais são os custos da Engenharia de Software?
 - Pesquisas mostram que 60% dos custos é para o desenvolvimento e 40% para os testes.
 - O custo de evolução do software, normalmente, excede o custo de desenvolvimento.
 - O custo depende do tipo de sistema a ser desenvolvido e suas restrições.
 - A distribuição dos custos depende do modelo de desenvolvimento adotado.

Conceitos

- Qual é a diferença entre a Engenharia de Software e a Ciência da Computação?
 - A Ciência da Computação tem como objetivo o desenvolvimento de teorias e fundamentações, enquanto que a Engenharia de Software se preocupa com as práticas de desenvolvimento de software.

Conceitos

- Qual é a diferença entre Engenharia de Software e Engenharia de Sistemas?
 - A Engenharia de Sistemas trata dos sistemas baseados em computadores, que inclui hardware e software. Enquanto a Engenharia de Software trata apenas dos aspectos de desenvolvimento de software.

Conceitos

- Princípios da Engenharia de Software
 - Formalidade: produtos mais confiáveis, controlar seu custo e mais confiança no seu desempenho.
 - Abstração: identificar os aspectos importantes, ignorando os detalhes.
 - Decomposição: subdividir o processo em atividades específicas, atribuídas a diferentes especialistas.
 - Generalização: sendo mais geral, é bem possível que a solução possa ser reutilizada.
 - Flexibilização: modificação com facilidade.

Conceitos

- A Engenharia de Software é uma disciplina que reúne:
 - Processos.
 - Métodos.
 - Ferramentas.
- Os processos seguem os métodos e estes se utilizam de ferramentas.
- Visam resolver problemas inerentes:
 - Ao processo.
 - Ao produto.

Conceitos

- Quais são os métodos da Engenharia de Software?
 - São abordagens estruturadas para o desenvolvimento de software que incluem os modelos de software, notações, regras e maneiras de desenvolvimento.

Conceitos

- O que são ferramentas CASE?
 - São programas de computador que têm o objetivo fornecer um suporte automatizado para as atividades de processo de software.
 - Operam em dois níveis:
 - Alto nível: ferramentas que suportam as atividades iniciais de requisitos e projetos.
 - Baixo nível: ferramentas que suportam as atividades de programação, depuração e testes.

Conceitos

- O que é um modelo de processo de software?
 - É uma representação simplificada de um processo de software, apresentada sobre uma perspectiva específica.
 - Os principais modelos são:
 - Cascata ou sequencial.
 - Modelo evolutivo.
 - Transformação formal.
 - Integração de componentes reusáveis.
 - Espiral.

Conceitos

- Modelos de processo
 - Especificam as atividades e a ordem em que, de acordo com o modelo, devem ser executadas.
 - Produtos de software podem ser construídos utilizando-se de diferentes modelos de processo.
 - Alguns modelos são mais adequados que outros para determinados tipos de aplicação.
 - A opção por um determinado modelo deve ser feita levando-se em consideração o produto a ser desenvolvido.

Conceitos

- Objetivos dos modelos
 - Auxiliar no processo de produção -> produtos de alta qualidade, produzidos mais rapidamente e a um custo cada vez menor.
 - Atributos: complexidade, visibilidade, aceitabilidade, confiabilidade, manutenibilidade, segurança etc.
 - Possibilitam:
 - Ao gerente: controlar o processo de desenvolvimento de sistemas de software.
 - Ao desenvolvedor: obter a base para produzir, de maneira eficiente, software que satisfaça os requisitos pré-estabelecidos.

Conceitos

- O que é um processo de software?
 - Um conjunto de atividades que objetivam o desenvolvimento e a evolução de software.
 - De maneira geral, as principais atividades são:
 - Especificação: define o que o sistema deverá fazer, considerando as suas restrições.
 - Desenvolvimento: produção do software.
 - Validação: checagem se o software faz o que o usuário deseja.
 - Evolução: mudanças no software para atender às novas demandas.

Processos de Desenvolvimento

- Definição:
 - Conjunto de atividades para especificar, projetar, implementar e testar sistemas de software.
 - As atividades necessárias para o desenvolvimento de software são:
 - Especificação.
 - Projeto.
 - Validação.
 - Evolução.

Processos de Desenvolvimento

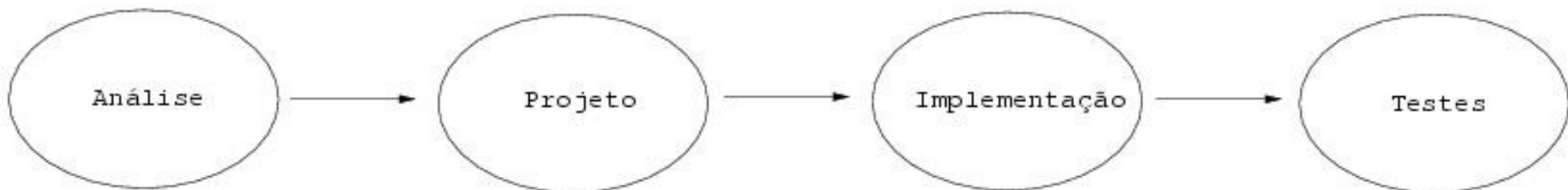
- Ciclo de vida clássico (Modelo Cascata)
 - Diferentes fases da especificação e desenvolvimento.
- Desenvolvimento evolutivo
 - Especificação e desenvolvimento são alternados.
- Desenvolvimento formal
 - Uso de modelo matemático é formalmente transformado em uma implementação.
- Desenvolvimento de sistemas baseado em reuso
 - O sistema é montado a partir de componentes.
- Desenvolvimento em espiral
 - Desenvolvimento evolutivo.

Ciclo de vida clássico

- Método sistemático e sequencial.
- O resultado de uma fase constitui na entrada de outra.
- Também é conhecido como cascata.
- Cada fase é estruturada como um conjunto de atividades que podem ser executadas por pessoas diferentes, simultaneamente.

Ciclo de vida clássico

- Fases:
 - Análise e definição de requisitos.
 - Projeto de software.
 - Implementação e teste unitário.
 - Integração e teste de sistema.
 - Operação e manutenção.



Ciclo de vida clássico

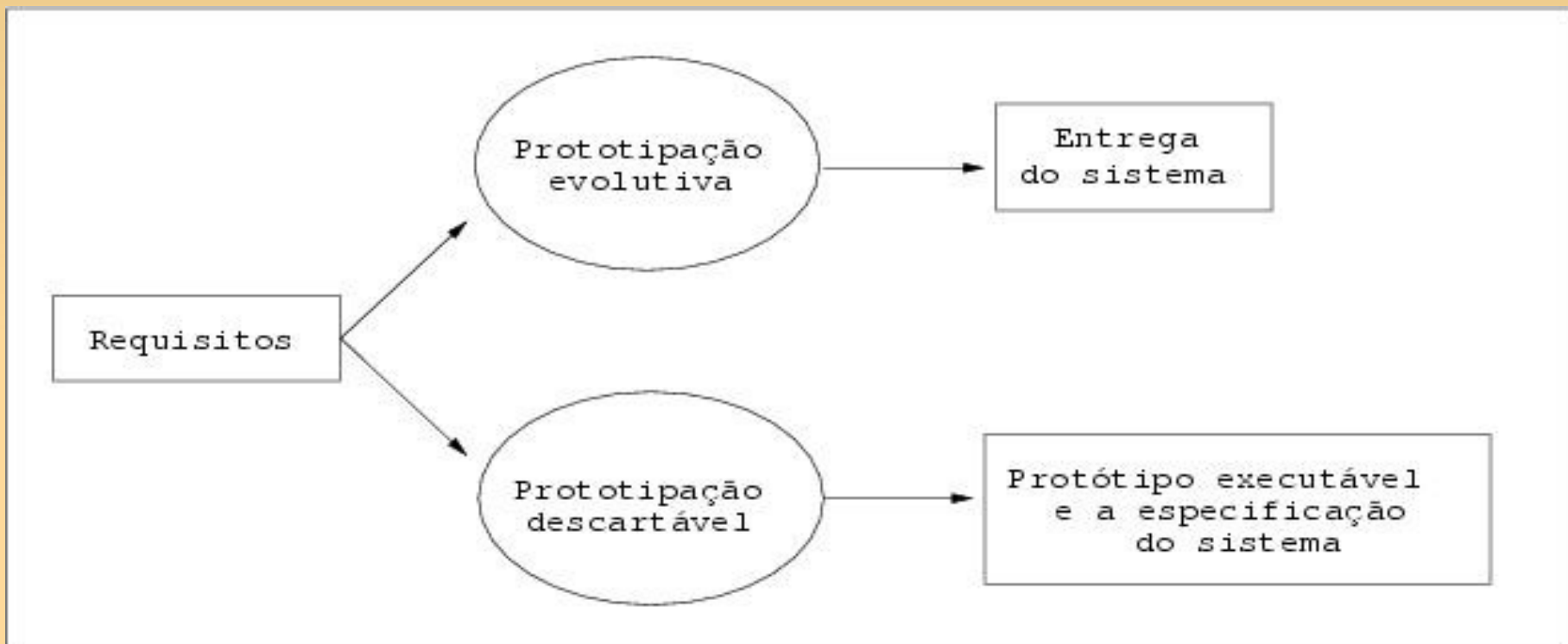
- Problemas:
 - Utiliza modelo sistemático e sequencial, em que a entrada de uma fase é o resultado da anterior.
 - O reinício do modelo é a dificuldade de acomodar mudanças depois que o processo está no final.
 - Dificuldade em atender às mudanças exigidas posteriormente pelo cliente.
 - Modelo mais adequado quando os requisitos estão muito bem entendidos.

Desenvolvimento Evolutivo

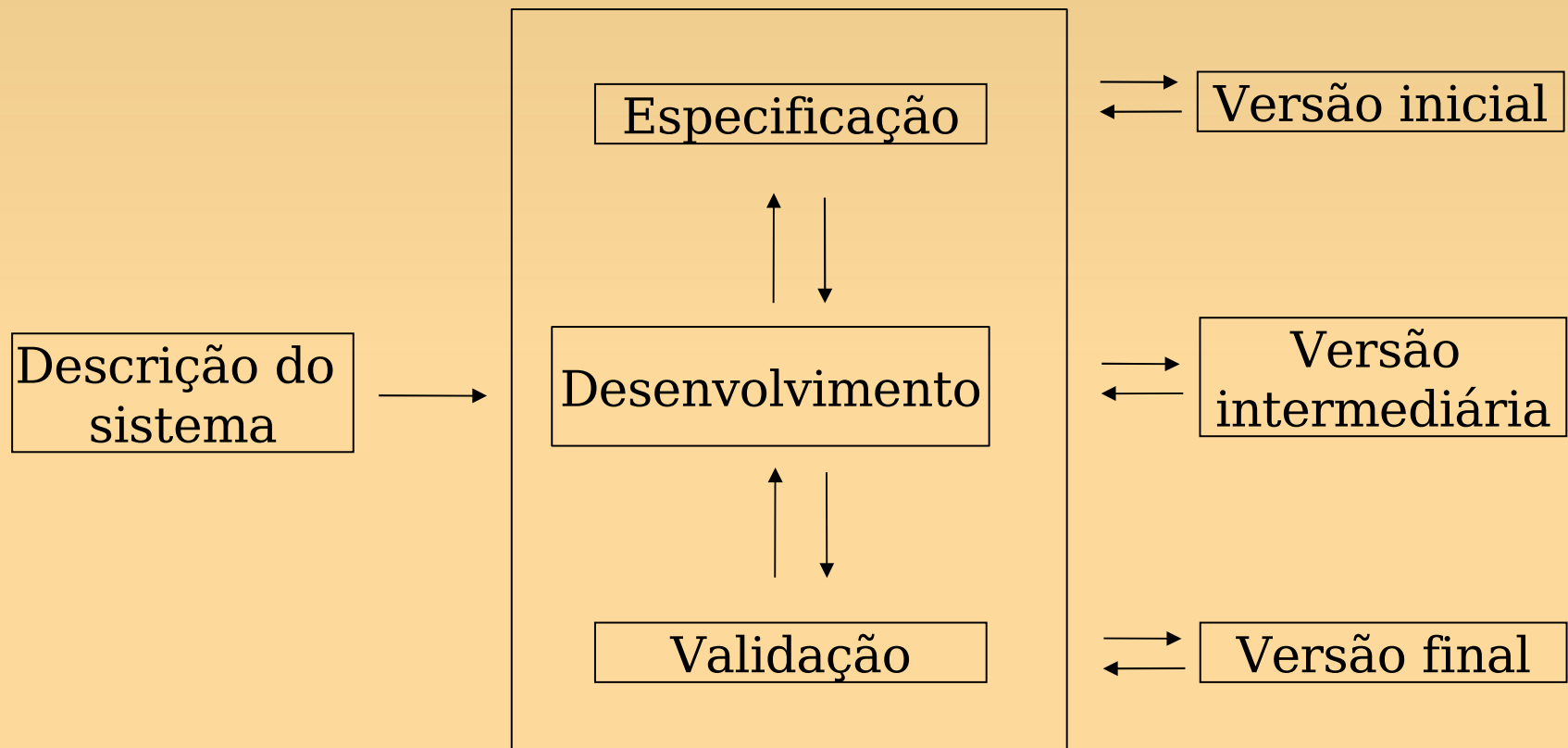
- Desenvolvimento Exploratório
 - O objetivo é desenvolver o sistema com o contínuo acompanhamento dos clientes desde da especificação. Os requisitos precisam ser bem entendidos.
- Prototipação Descartável
 - O objetivo é entender os requisitos do sistema.

Desenvolvimento Evolutivo

- Prototipação



Desenvolvimento Evolutivo

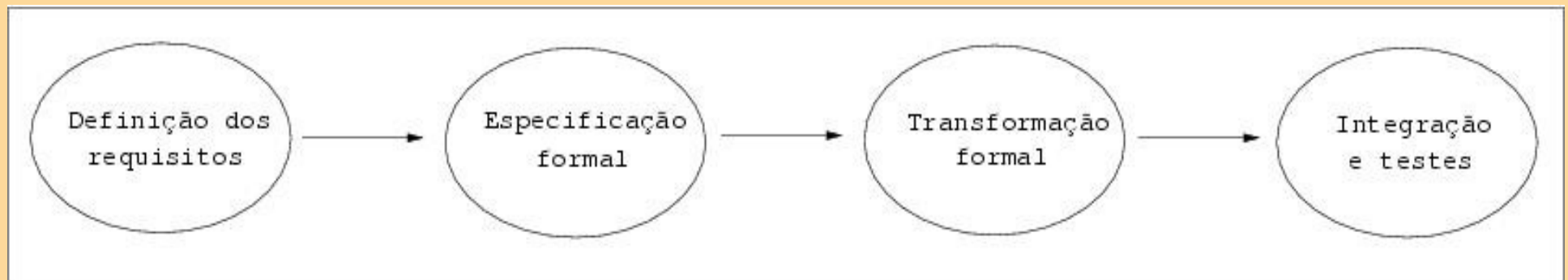


Desenvolvimento Evolutivo

- Problemas:
 - Ausência de visibilidade do processo.
 - Sistemas são fracamente estruturados.
 - Necessidades de ferramentas de rápido desenvolvimento.
- Aplicabilidade:
 - Sistemas de pequeno e médio porte.
 - Como parte de um sistema grande.
 - Sistema de curta duração.

Desenvolvimento Formal de Sistemas

- Baseado na transformação de uma especificação matemática, através de diferentes representações para um programa executável.
- Conseguir alcançar os requisitos da especificação mais facilmente.



Desenvolvimento Formal de Sistemas

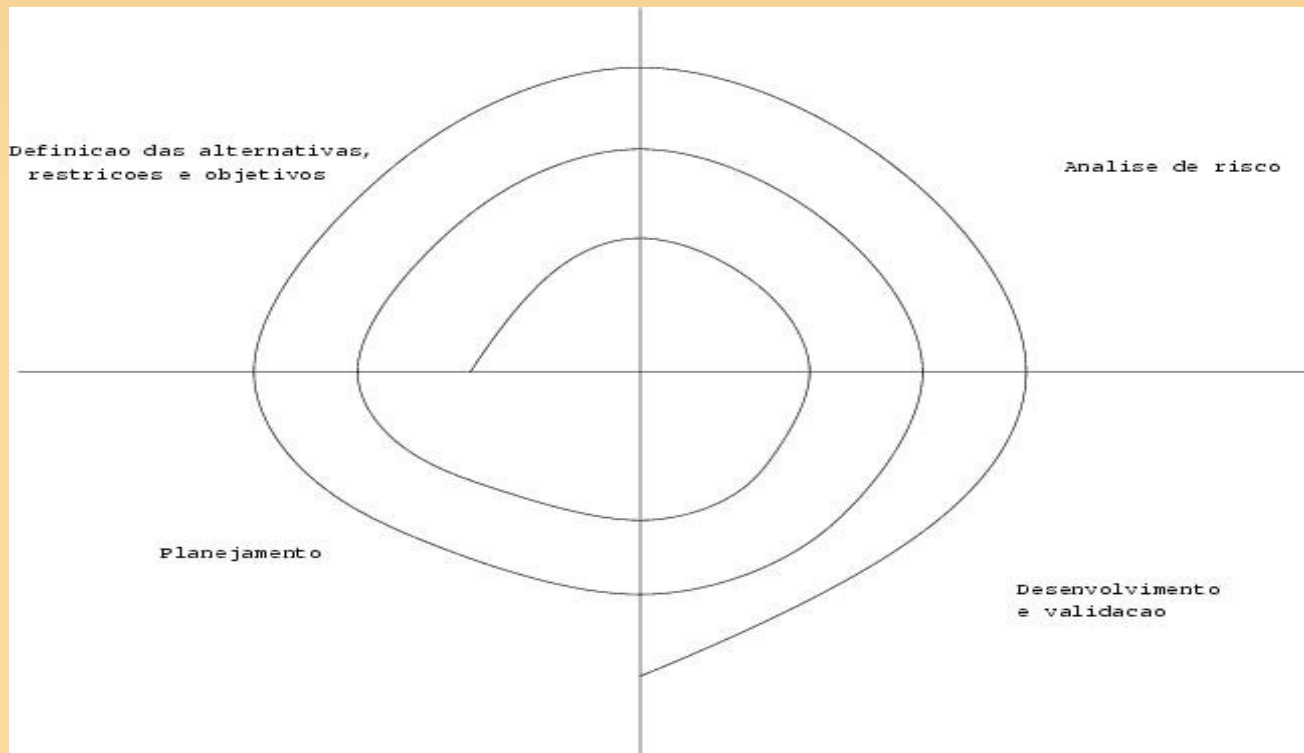
- Problemas:
 - Dificuldade em encontrar profissionais especializados.
 - Dificuldade em especificar determinados aspectos como a interface do usuário.
- Aplicabilidade:
 - Principalmente para sistemas críticos, onde não são toleradas falhas.

Desenvolvimento baseado em reuso

- Os sistemas são baseado em componentes já existentes, semelhantes ao desenvolvimento de hardware.
- Fases do processo:
 - Análise do componente.
 - Modificação dos requisitos.
 - Projeto do sistema com reuso.
 - Desenvolvimento e integração.
- Método que vem crescendo bastante nos últimos tempos.

Modelo Espiral

- Engloba as características do ciclo de vida clássico e evolutivo, ao mesmo tempo que adiciona um novo elemento – a análise de risco.



Modelo Espiral

- Fases
 - Definição dos objetivos, alternativas e restrições.
 - Análise de risco.
 - Desenvolvimento e validação.
 - Planejamento.

Conceitos

- Quais são as questões atuais?
 - Sistemas legados: os sistemas antigos devem ser mantidos e atualizados.
 - Heterogeneidade: sistemas são uma combinação de hardware e software.
 - Prazos de entrega: pressão para um menor prazo de entrega.

Principais aspectos da aula

- A Engenharia de Software é uma disciplina da Engenharia que se preocupa com todos os aspectos da produção de software.
- Produtos de software consistem no desenvolvimento de programas e documentação associada. Manutenibilidade, independência, validação e evolução são os atributos essenciais.

Principais aspectos da aula

- O processo de software consiste de atividades que são envolvidas no desenvolvimento de produtos de software. As atividades básicas deste desenvolvimento são: especificação, desenvolvimento, validação e evolução.
- Métodos são formas organizadas de produzir software.

Principais aspectos da aula

- Ferramentas CASE são programas que são projetados para auxiliar nas atividades rotineiras no processo de desenvolvimento de software, tais como: desenho de diagramas, checagem da consistência dos diagramas e manutenção do programa.

Questões de revisão

- Qual é a importância da Engenharia de Software no desenvolvimento de sistemas?
- Faça um estudo comparativo entre o modelo de desenvolvimento de software tradicional e o evolutivo.
- Comente os princípios da Engenharia de Software.
- Qual é a importância do software no cotidiano das pessoas?

Referências Bibliográficas

- Carvalho, Ariadne M. B. Rizzoni & Chiossi, Thelma C. dos Santos. *Introdução à Engenharia de Software*. Unicamp, 2001.
- Paulo Filho, Wilson de Pádua. *Engenharia de Software*. LTC, 2003.
- Pressaman, Roger S. *Engenharia de Software*. 6a edição. McGraw-Hill, 2006.
- Sommerville, Ian. *Engenharia de Software*. 8a edição. Pearson Education, 2007.