Slide 1- principios fundamentais  
Software é projetado criado e mantido, engloba programas documentos e estrutura de dados.  
Software é um elemento de sistema lógico, não é “fabricado”, é desenvolvido, se deteriora(precisa de updates), quando melhor o projeto do software, mais difícil ele se deteriorar.  
Deve ser projetado e implementado para o REUSO  
Modificações custam tempo e dinheiro. Acontecem por falta de engenharia de software.  
Categoria de Softwares: de sistema - de aplicações – científico e de engenharia – embutido – comercial – da web – inteligência artificial

Engenharia de software veio para responder a crise de software, melhora e organiza os processos de criação de software.

Software Legado – são softwares antigos que são modificados para atender necessidades atuais, modificações dificultam sua existência pois, geralmente não foram documentados ou se foram está ruim.  
Modificações em softwares occorem quando: o cliente solicita coisa nova, adaptação a novo ambiente, correção de erros.  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
Para alta qualidade, segue-se um (roteiro)=processo de software – engenheiros adaptam as suas necessidades, quem solicitou define constrói e testa. Isso fornece controle.

UML – UNIVERSAL MODELING LANGUAGE

Processo de Desenvolvimento de Software  
Ferramentas: Apoio automatizado aos métodos CASE  
Métodos: Como fazer - Comunicação, Análise de Requisitos, Modelagem de Projeto, Construção de programação, Testes, Manutenção.   
Processos: Elo entre ligação e ferramentas, é a base do controle gerencial, estabelece onde os métodos serão aplicados, o que será produzido: modelos, documentos, relatórios, etc., assegurar a qualidade e coordenar alterações, marcar progresso.  
Foco na qualidade: Compromisso organizacional: processo eficiente, definido, gerenciado, medido e controlado.

Atividade Guarda chuvas são atividades que serão realizadas durante todo o processo.  
Exemplo: Controle de projeto, gestão de risco, garantia de qualidade, revisão, gestão de reuso, etc.

Estrutura comum de processo – Alicerce para um processo completo, identifica atividades de estrutura que são aplicados em qualquer projeto, cada atividade é preenchida por um conjunto de ações de engenharia de software, por exemplo: levantamento de requisitos  
Atividades de Estrutura: Comunica, planeja, modela, constrói, implanta. –Auto Explicativo.

Exemplos de falha de processo: Atraso, custo crescente, má funcionamento.

CMMI – Capacity Maturity Modeling Integration – Mede empresas de criação de software.  
Eng de software apoia nos processos e no gerenciamento  
Níveis -1:Inicial 2:Gerido 3:Definido 4:Quantitativamente Gerido 5:Otimizado  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Modelos de Processo de Software – Serve para generalizar a estrutura de processos.

Modelo de Cascata – Mais antigo, Entrega tudo de uma vez. Problemas: Raramente segue o fluxo sequencial que o modelo propõe, difícil o cliente especificar tudo que quer, executável só no final do projeto.  
Modelo Incremental – Combina elementos de cascata repetidamente: cliente recebe versão picada, fala pra adicionar coisas e repete o processo. Entrega picada. Importante quando o os requisitos não são explicitamente detalhados. Apropriado para sistemas pequenos, seus testes são mais efeitos.  
Modelo RAD – Cascata + Incremental. Desenvolvimento rápido. Exige compromentimento  
Prototipagem é gerar produtos não finalizados para análise do cliente e melhora do software. Problemas: qualidade global, implementações comprometidas, cliente e desenvolvedor discordarem que o protótipo é construído apenas para definir requisitos.  
Modelo Espiral – Prototipagem + Cascata . é um ciclo com preocupação de metodologia. Para desenvolvimento de grande Softwares custo e cronograma baseados no feedback do cliente.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Diagrama caso de Uso – Apresentar visão geral das funções e serviços que o sistema deverá fornecer sem se preocupar com o COMO  
Casos de Uso: tarefas – balãozinho  
Atores: Quem vai usar

Generalização: associação entre Cus com características semelhantes  
Include: Obrigatoriamente.  
Extend: Só ocorre se uma condição for satisfeita.