

Análise de Sistemas – AULA 02

BCC Noturno - EMA908915A



Prof. Rafael Oliveira
rpaes@ic.unesp.br

Universidade Estadual Paulista
“Júlio de Mesquita Filho”
UNESP

Rio Claro 2014 (Sem 2)

- Disciplina:
 - Objetivos;
 - Materiais e Métodos;
 - Critérios de avaliação; e
 - Datas importantes.

- O que são sistemas?
- Tipos de sistemas;
- Papel do analista de sistemas;

- Conceitos importantes:
 - Engenharia de Software
 - Análise de Sistemas;
 - Processo de Software;
- Modelos Prescritivos de Software (Ciclos de Vida de Software);

- Software:
 - O que é?
 - Quem faz?
 - Por que é importante?
 - Quais são os passos?
 - O que este trabalho produz?

- Software:
 - **O que é?**
 - Quem faz?
 - Por que é importante?
 - Quais são os passos?
 - O que este trabalho produz?

Software e Engenharia de Software

- Software: **O que é?**
 - Um software é um produto que profissionais da área de software é um constroem e matêm a longo prazo;
 - O software engloba programas que executam em um computador com alguma arquitetura específica;
 - A **Engenharia de Software** engloba um processo, uma coleção de métodos (práticas) e um conjunto de ferramentas que permitem que profissionais contruam software de qualidade.

- Software: **O que é?**

Pressman 2010

- Software is: (1) instructions (computer programs) that when executed provide desired features, function, and performance; (2) data structures that enable the programs to adequately manipulate information, and (3) descriptive information in both hard copy and virtual forms that describes the operation and use of the programs.

- Software:
 - O que é?
 - **Quem faz?**
 - Por que é importante?
 - Quais são os passos?
 - O que este trabalho produz?

- Software: **Quem faz?**
 - Engenheiros de Software constroem e suportam (apoiam) o software a longo prazo;
 - Virtualmente, qualquer pessoa, virtualmente, em um mundo industrializado virtual usa esse software diretamente ou indiretamente.

- Software:
 - O que é?
 - Quem faz?
 - **Por que é importante?**
 - Quais são os passos?
 - O que este trabalho produz?

- Software: **Por que é importante?**
 - O **Software** é importante porque ele afeta intimamente todos os aspectos de nossas vidas e se tornaram pervasivos para operações de comércio, nossa cultura, e nossas atividades cotidianas.
 - A **Engenharia de Software** é importante por que ela propicia a nós a construção de sistemas complexos em um tempo viável e com alta qualidade.

- Software:
 - O que é?
 - Quem faz?
 - Por que é importante?
 - **Quais são os passos?**
 - O que este trabalho produz?

- Software: **Quais são os passos?**
 - Devemos produzir software como outros produtos bem sucedidos são produzidos;
 - Por meio da aplicação de processos ágeis e adaptáveis que levem a resultados de alta qualidade que atingem as necessidades dos clientes que irão consumir tal software;
 - Sendo assim, desenvolvedores devem aplicar abordagens de engenharia.

- Software:
 - O que é?
 - Quem faz?
 - Por que é importante?
 - Quais são os passos?
 - **O que este trabalho produz?**

Software e Engenharia de Software

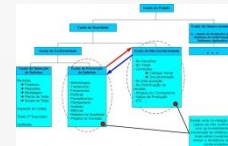
- Software: **O que este trabalho produz?**
 - Do ponto de vista de um Engenheiro de Software, o produto de software é um conjunto de programas, conteúdos (dados) e outros produtos que configuram o produto final;
 - Entretanto, do ponto do usuário, o produto de software é a informação resultante que de algum modo deixa o mundo do usuário melhor.

- Contexto:

Cliente



**Equipe de
Desenvolvimento
de Software**



- **Contexto:**

Décadas de 60 e 70:

Desenvolver hardware que reduzisse o custo de processamento e de armazenamento de dados

Década de 80:

Avanços na microeletrônica resultaram em um aumento do poder computacional a um custo cada vez menor

Projetos de desenvolvimento de software, deixavam muito a desejar: cronogramas não eram cumpridos, custos excediam os previstos, o software não cumpria os requisitos estipulados

Últimas duas décadas:

Desafio: melhorar a qualidade e reduzir o custo do software produzido, através da introdução da disciplina no desenvolvimento, o que é conhecido como engenharia de software.

Definição dada pela IEEE [IEE93]:

- Aplicação de uma abordagem sistemática, disciplinada, e quantificável, para o desenvolvimento, operação e manutenção de software.
- O estudo de abordagens sistemáticas, disciplinadas e quantificáveis.

Software como um Produto;

A Engenharia de Software como criadora do Produto (**Técnicas, Metodologias, Ferramentas que agilizem o desenvolvimento/manutenção de software**);

Software (Concepção – Manutenção);

Na sociedade moderna, o papel da engenharia é fornecer sistemas e produtos que melhoram os aspectos materiais da vida humana, tornando assim a vida mais fácil, menos perigosa, mais segura e mais agradável.

- **Sistemas:**

- Desempenham papel preponderante no dia a dia das pessoas
- Há aplicações extremamente críticas
 - Espaçonaves, aeronaves, trens, carros, reatores nucleares e equipamentos hospitalares
- Devem funcionar corretamente
- Devem ser totalmente confiáveis
 - Devem cumprir seus requisitos integralmente

- **Análise de Sistemas:**

- Atividade Complexa;
- Variabilidade de informações;
- Descrição adequada das interações entre o software e o ambiente;

1º Passo:

- Obtenção das informações relevantes à aplicação
- Análise destas informações para determinação do domínio do problema



- **Como contornar a complexidade do software ?**
 - Surgimento de novas áreas de aplicação
 - Exigência de maior confiabilidade e precisão dos sistemas
- **Qual a melhor prática para o desenvolvimento de sistemas de software?**
 - Grande esforço para encontrar soluções para resolver as dificuldades oriundas da:
 - própria natureza abstrata do software
 - sua complexidade
 - necessidade de cumprir seus objetivos
 - rapidez com que pode sofrer alterações

A **ANÁLISE DE SISTEMAS** é focada para as etapas iniciais do processo de desenvolvimento dos sistemas de software:

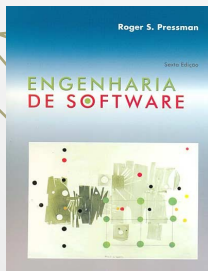
- Análise e Especificação dos requisitos
- **Importante :** Etapas Cruciais para o desenvolvimento do sistema de software para que se cumpram os seus requisitos de maneira plenamente satisfatória.

Processo de software

- Visão Genérica:



desenvolvimento



Técnicas de ES

+



Ferramentas



Software

Processo de software



- Visão Genérica:
 - Um processo é uma coleção de atividades, ações e tarefas que são realizadas quando algum produto deve ser criado.
 - Uma atividade se esforça para alcançar um amplo objetivo (ex. Comunicação com stakeholders) e é aplicado independentemente do domínio da aplicação, tamanho do projeto, complexidade de esforços, ou grau de rigor com o qual a engenharia de software é aplicada.
 - Uma ação (ex: projeto arquitetural) engloba um conjunto de tarefas que produz um grande produto (ex: um modelo de projeto arquitetural).
 - Uma tarefa foca em um pequena, mas com um objetivo bem-definido (ex: conduzindo uma unidade de teste) que produz um resultado tangível.

- Um **stakeholder** é alguém que tem interesse no resultado bem sucedido do resultado do projeto:
 - Gerentes de negócio;
 - usuários finais;
 - engenheiros de software;
 - Pessoas de suporte, etc.

- Conceitos de Processo de Software:
 - Segundo **Sommerville (2008)**, as empresas que obtiveram sucesso no desenvolvimento de software estão utilizando **processos de software definidos**. Em linhas gerais, um processo de software pode ser definido como um conjunto de etapas que devem ser seguidas para o desenvolvimento de um produto.

- Conceitos de Processo de Software:
 - Segundo **Pressman (2005, 2006)**, processo de software, é um conjunto de passos constituindo um guia, que ajuda na execução destes com **garantia de qualidade** e com **prazos estabelecidos**. Sua importância está centralizada no controle de execução das atividades estabelecidas e, dessa forma, facilitando a execução do trabalho em equipe.

- Conceitos de Processo de Software:
 - Diante dos conceitos apresentados sobre processo de software é necessário comentar sobre os **modelos de processos de software** existentes. Na literatura podem ser encontrados diversos modelos: modelo em cascata, desenvolvimento evolucionário, transformação formal, montagem de um sistema a partir de componentes reutilizáveis, entre outros (SOMMERVILLE, 2008) (PFLEEGER, 2004) (PRESSMAN, 2005) (PRESSMAN, 2006).

Processo de software



- Conceitos de Processo de Software:
 - Os autores mencionam também a importância de ferramentas CASE, *Computer Aided Software Engineering*, na execução das atividades para automatização das etapas de documentação, modelagem, desenvolvimento e teste.

Se o processo for fraco, o produto final sofrerá inevitavelmente, mas uma ênfase exagerada também é perigosa.

Processo de software



- No contexto de Engenharia de Software, um processo não é uma prescrição rígida para como construir um software.
- Ao contrário, um processo deve ser uma abordagem adaptável que possibilita que as pessoas envolvidas no trabalho de desenvolvimento (o time do software) escolham um conjunto apropriado de ações e tarefas;
- A intenção é sempre entregar um software em tempo aceitável de projeto e com qualidade suficiente para satisfazer aqueles que patrocinaram sua criação e irão utilizá-lo posteriormente.

Processo de software



- Um framework de um processo estabelece o fundamento para um processo de engenharia de software completo por meio da adaptação de um pequeno número de atividades que são aplicáveis a qualquer projeto de software, independente de seu tamanho ou complexidade.
- Adicionalmente, o processo engloba um conjunto de atividades de guarda-chuva que são aplicáveis transversalmente em todo o processo de software.
- Atv. Guarda-chuva: Atividades que servem de apoio às atividades de um framework (arcabouço).

Processo de software



- Um framework genérico de engenharia de software engloba 5 atividades:
 - Comunicação;
 - Planejamento;
 - Modelagem;
 - Construção;
 - Implantação.

Processo de software



- Comunicação:
 - Antes de alguma atividade técnica ser iniciada, é criticamente importante comunicar e colaborar com o cliente (e outros stakeholders);
 - A intenção é entender os objetivos dos stakeholders para o projeto e reunir requisitos para ajudar a definir características e funções do software.

- Planejamento:
 - Qualquer jornada complexa pode ser simplificada se um mapa existe.
 - Um projeto de software é uma jornada complexa, e o planejamento da atividade cria uma mapa que ajuda a guiar o time a traçar uma jornada.
 - Tal mapa, chamado de plano de projeto de software, define o trabalho de engenharia de software por meio da descrição de tarefas a serem conduzidas, os prováveis riscos, os recursos a serem requeridos, os produtos a serem produzidos e o cronograma.

- Modelagem:
 - Se você é um paisagista, um construtor de pontes, um aeronáutica engenheiro, um carpinteiro, ou um arquiteto, você trabalha com modelos de todos os dias.
 - Você cria um "esboço" da coisa, de modo que você vai entender o cenário ampliado que ela vai parecer arquitetonicamente, como as partes constituintes se encaixam, e muitas outras características.
 - Se necessário, você refina o esboço para maiores detalhes, como forma de um esforço para entender melhor o problema e como você irá resolvê-lo.
 - Um engenheiro de software faz a mesma coisa com a criação de modelos para entender melhor os requisitos de software e do projeto que vai implementar esses requisitos.

Processo de software



- Construção:
 - Esta atividade combina geração de código (manual ou automatizada) e o teste que é requerido para descobrir erros em código.

Processo de software



- Implantação:
 - O software (como uma entidade completa ou como um incremento parcialmente completo) é entregue para o cliente que avalia o produto recebido e provê um feedback baseado na avaliação.

Processo de software



- Estes 5 (cinco) arcabouços (frameworks) genéricos de atividades podem ser utilizados durante o desenvolvimento de:
 - um pequeno sistema;
 - Programa simples;
 - Criações de grandes aplicações Web;
 - Engenharia de grandes e complexos sistemas

Os detalhes do processo de software serão um tanto quanto diferentes em cada caso, mas o arcabouço de atividades continua sendo o mesmo.

Processo de software



- Para muitos projetos de software, as atividades do framework são aplicadas interativamente como um projeto progressivo.
- Isto é, comunicação, planejamento, modelagem, construção e implantação são aplicados repetidamente através de um número de iterações do projeto;
- Cada iteração de projeto produz um incremento do software que provém aos stakeholders um conjunto de novas características e funcionalidades.
- Assim que cada incremento é produzido, o software se torna mais e mais completo.

Processo de software



- Atividades de arcabouços (frameworks) de processos de engenharia de software são complementadas por um número de atividades de guarda-chuva.
- Em geral, atividades de guarda-chuva são aplicadas através de um projeto de software e ajudam a gerenciar e controlar o progresso, qualidade, mudanças e riscos do software.

Processo de software



- Atividades típicas de guarda-chuva incluem:
 - Acompanhamento de projetos e controle:
 - Gestão de riscos
 - Garantia de qualidade do software:
 - Revisões técnicas;
 - Métricas/Medições;
 - Gestão de configurações de software;
 - Gestão de reusabilidade;
 - Preparação e produção do produto.

- **Acompanhamento de projetos e controle:**
 - permite que o planejamento de projeto e a tomada de alguma decisão necessária para manter o cronograma.

- **Gestão de riscos:—**

- Acessa os riscos que podem afetar os resultados do projeto ou a qualidade do produto.

- **Garantia da qualidade do software:—**
 - Define e conduz atividades requeridas para garantir a qualidade do software em desenvolvimento.

- Typical umbrella activities include:
 - **Revisões técnicas—**
 - Acessa a produto da engenharia de software em um esforço para descobrir ou remover erros antes de que eles sejam propagados para uma próxima atividade.

- **Métricas/Medições:**—define e coleta processo, projeto e métricas do produto de software que ajudam o time na entrega de um resultado que alcance as necessidades dos stakeholders; Podem ser utilizados em conjunto com outros frameworks e atividades de guarda-chuva.

- **Gestão de configurações de software;**—
gerencia efeitos de mudanças e alterações genéricas e específicas dentro de módulos e do sistemas como um todo através do processo de software.

- **Gestão de reusabilidade**— Define critérios para esforços na identificação de reusos (incluindo componentes de software) e estabelece mecanismos para alcançar componentes reutilizáveis.

Processo de software



- Fases de um Processo de Desenvolvimento de Software **(Linhas Gerais)**:
 - Análise de requisitos e Especificação do Problema;
 - Projeto de Software;
 - Implementação;
 - Verificação e Validação;
 - Manutenção.
- Ciclos de Vida de Software:
 - Modelos Prescritivos: prescrevem como algo deve ser realizado;
 - Modelos ágeis

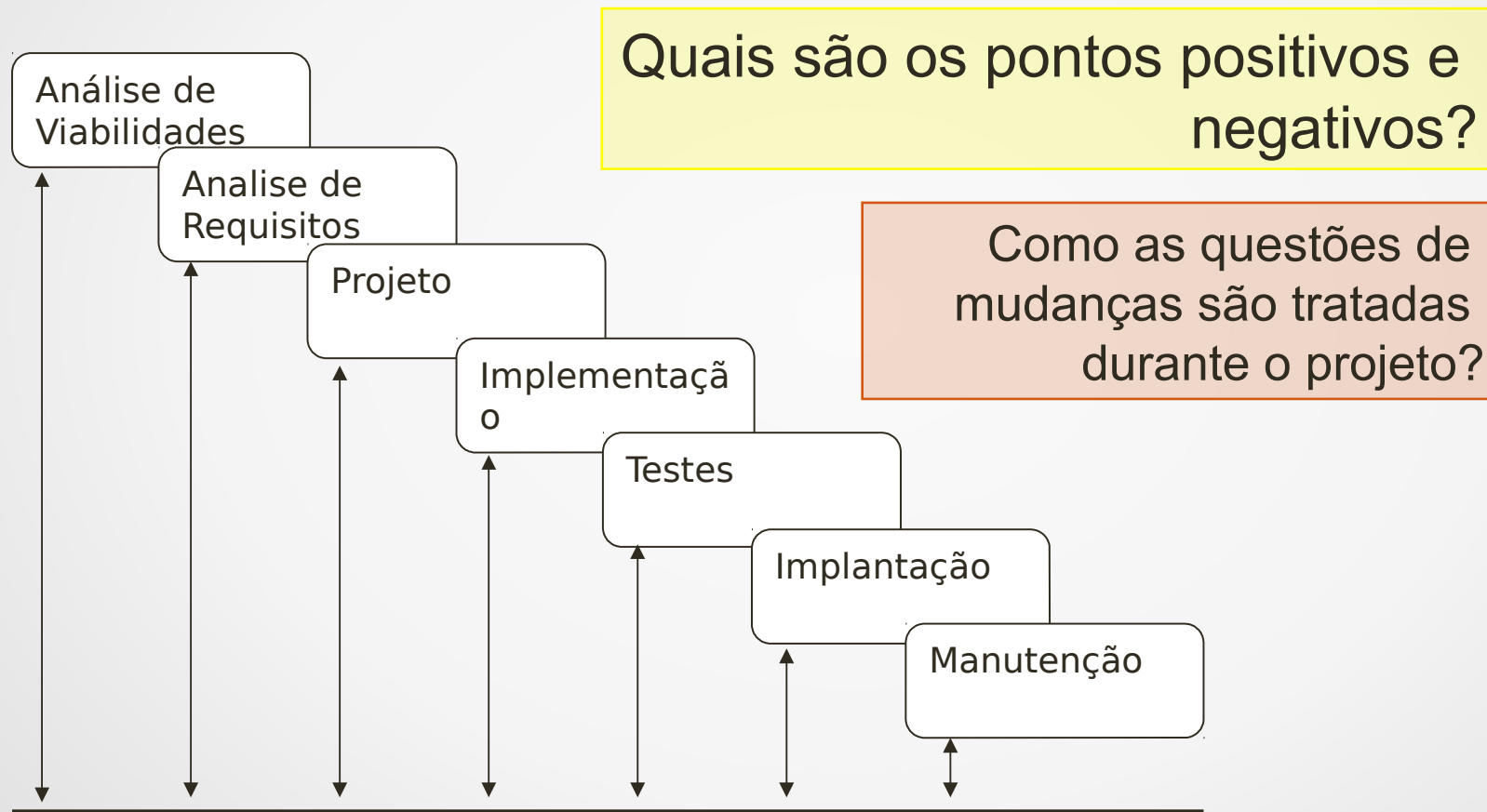
Modelos de processo



- **Modelos prescritivos:** Enfatizam a definição, identificação e aplicação detalhada de atividades e tarefas de processo (Modelo em Espiral).
- **Modelos ágeis:** Enfatizam a agilidade do projeto utilizando abordagens mais informais, enfatizando manobrabilidade e adaptabilidade (Scrum).
- Mais detalhes na próxima AULA!!!!

Modelos de processo

- **Ciclo de Vida: Sequencial ou Cascata:**



Modelos de processo



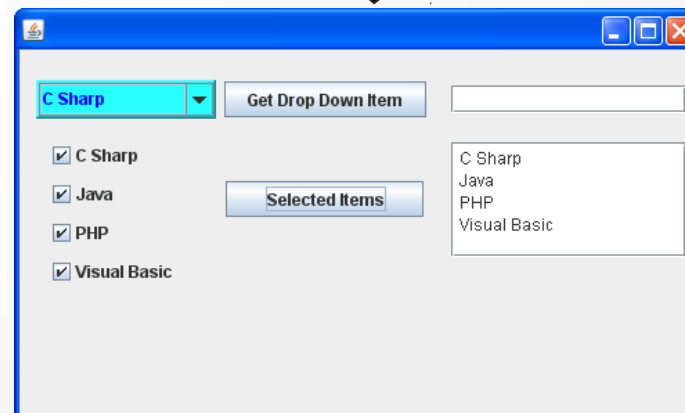
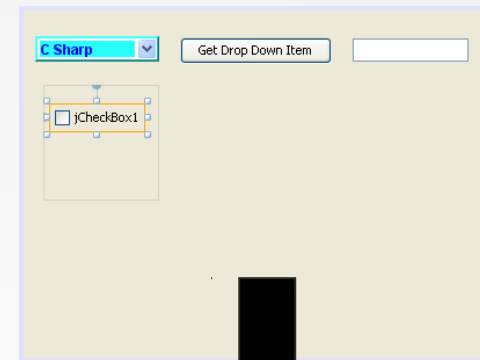
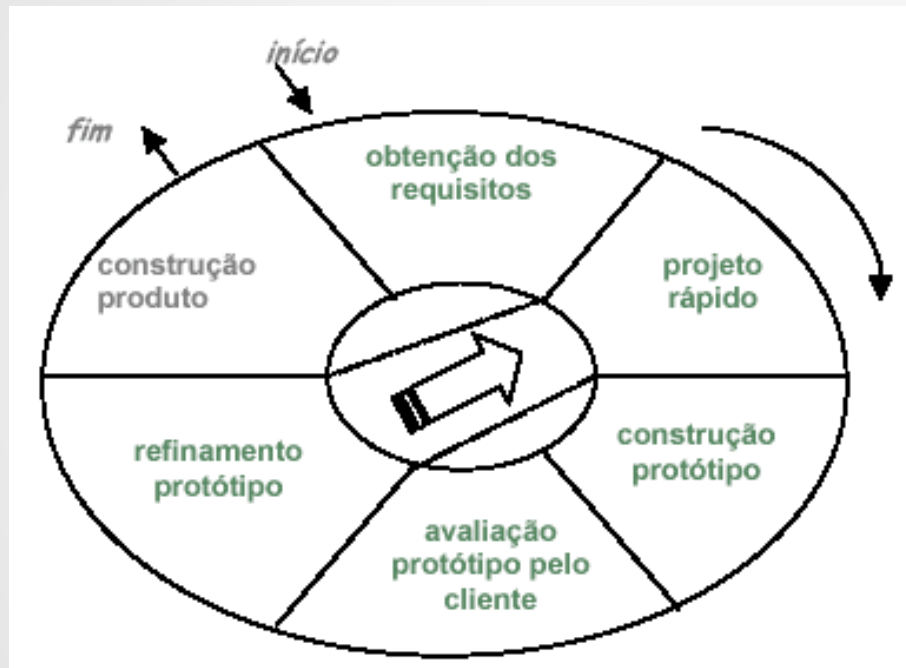
Modelos Evolucionários de Processo: são modelos **ITERATIVOS**, permitindo que os engenheiros de software criem versões cada vez mais completas do software;

Modelos:

Modelo Prototipagem: projeto com definição de objetivos gerais; não possui detalhes; alto grau de incertezas; melhor comunicação com o cliente;

Modelos de processo

- Modelo em Prototipagem:



Questões importantes?!?!?

- EXERCÍCIOS

Exercício 01



- Pensar como CLIENTE em um sistema de software;
- Exemplos:
 - Locadora de Vídeo;
 - Locadora de Carros;
 - Consulta Médica;
 - Controle de Estoque;
 - Biblioteca;
 - Imobiliária;
 - Gestão escolar
 - Livraria
 - Etc.

Exercício 01



- Escolher 1 dos sistemas mencionas no slide anterior e idealizar tal sistema. Exercite sua criatividade e crie um documento descrevendo o seu sistema. Tal documento deve conter, entre outros tópicos:
 - Qual o “problema” a ser solucionado pelo sistema;
 - Por que o sistema é necessário?
 - Quais as principais funcionalidades!
 - Há algum relatório a ser gerado?
 - Quantos tipos e quais os usuários a interagir com o sistema? (ex: adm e funcionario)
 - Como você deseja o sistema? Telas, visões, performance.
 - Há necessidade de alguma especificação particular? (Mala direta, suporte, etc.)
 - Em quais/quantos canais o sistemas deve funcionar? (Mobile, Web, Local etc.)
 - Outros detalhes ...

Exercício 02



- Your company Softa Inc. has got four projects. Suggest the most appropriate generic software process model which might be used as a basis for managing the development of the following systems:
 - a system to control anti-lock braking in a car,
 - a next generation computer game,
 - a library book lending system that replaces an older version, and
 - a software to monitor stock market.

Exercício 03



- You have been appointed a project manager within an information systems organization. Your job is to build an application that is quite similar to others that your team has built, although this one is larger and more complex. Requirements have been thoroughly documented by the customer. What software process model would you choose and why?

Exercício 04



- Umbrella activities are common to all software process models. A typical example of an umbrella activity is documenting. Documentation exists regardless of the chosen process model and activity. What other umbrella activities can you think of? Do you think that we can bend from the umbrella activities and thus save time in an otherwise late project?

Exercício 05

- Analise criticamente a seguinte frase baseando-se no conteúdo da disciplina:
- “The best programmers do not always make the best software managers”