

Engenharia de Software I

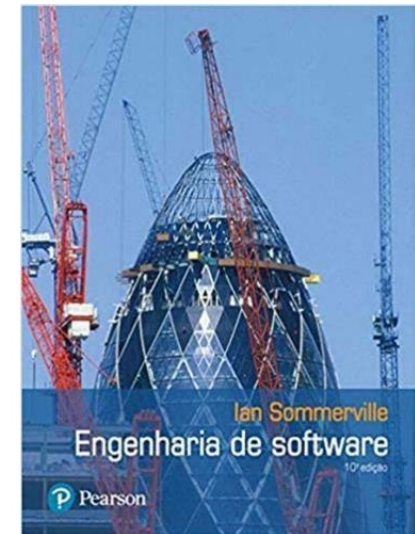
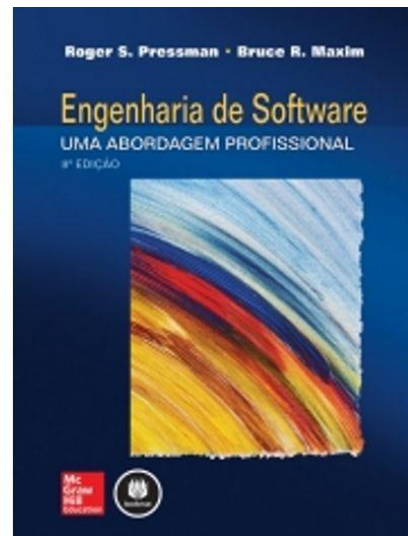
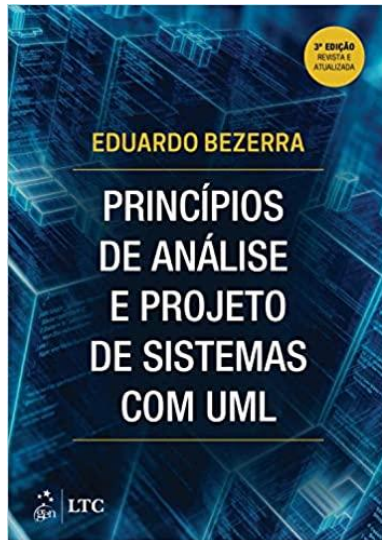
Profº Ramon Trigo

Objetivos da Disciplina

- Identificar as características de Sistemas de Informação, seus tipos, viabilidade técnica, características de custo, valor e qualidade da informação.
- Explicar as características de um sistema, seus componentes e relacionamentos.
- Compreender o ciclo de vida utilizando concepções do modelo cascata.
- Utilizar conceitos da UML na análise de requisitos e na elaboração de diagramas focando na modelagem de sistemas.

Ementa

- Introdução à Análise de Sistemas. Modelos de Ciclo de Vida de Software.
- Modelos de Processos de Desenvolvimento de Software (Modelo em Cascata, Espiral e Prototipagem).
- Definição e classificação de Requisitos de Software (funcionais e não funcionais). Técnicas de Levantamento de Requisitos.
- Modelo de Negócios aplicado ao levantamento de Requisitos (Canvas).
- Estudo de Viabilidade.
- Técnicas de documentação.
- Metodologias para desenvolvimento de sistemas



Conteúdo

OBJETIVOS

- ✓ Compreender o que é a engenharia de software e porque ela é importante;
- ✓ Desenvolvimento dos diferentes tipos de sistema de software pode exigir diferentes técnicas de engenharia;
- ✓ Questões éticas e profissionais importantes para os engenheiros de software;

Introdução

O mundo moderno não funciona sem software. A infraestrutura e os serviços públicos nacionais são controlados por sistemas computacionais e a maioria dos produtos elétricos inclui um computador e um software que o controla.

A produção e distribuição industriais são completamente informatizadas, assim como o sistema financeiro.

O setor de entretenimento — incluindo a música, os jogos de computador, o cinema e a televisão, usa software intensivamente.

Mais de 75% da população mundial possui um telefone celular controlado por software; e, até 2016, quase todos eles acessarão a internet.

Isso simplifica a engenharia de software, uma vez que não há limites naturais para o potencial do software.

No entanto, devido à ausência de limitações físicas, os sistemas de software podem rapidamente se tornar bastante complexos, difíceis de entender e caros de modificar.

Não há notações, métodos ou técnicas universais para a engenharia de software, pois os diferentes tipos de software exigem abordagens igualmente diferentes.

Desenvolver um sistema de informações organizacionais é completamente diferente de desenvolver um sistema de estoque.

Nenhum deles têm muita coisa em comum com um jogo de computador que utiliza os gráficos intensivamente.

Todas essas aplicações precisam de engenharia de software, mas nem todas elas precisam dos mesmos métodos e das mesmas técnicas.

A engenharia de software é criticada como inadequada para o desenvolvimento de software moderno.

Entretanto, acredito que muitas dessas ditas falhas de software sejam consequência de dois fatores:

Complexidade crescente dos sistemas. À medida que as novas técnicas de engenharia de software nos ajudam a construir sistemas maiores e mais complexos, as demandas mudam. Os sistemas precisam ser construídos e distribuídos com mais rapidez; é necessário que sejam ainda maiores e mais complexos; e eles precisam ter novas capacidades, que antes eram consideradas impossíveis.

Não utilização de métodos de engenharia de software. É bem fácil escrever programas de computador sem usar métodos e técnicas de engenharia de software. Muitas empresas se aventuraram no desenvolvimento de software conforme seus produtos e serviços evoluíram. Como não usam métodos de engenharia de software em seu cotidiano, seu software costuma ser mais caro e menos confiável do que deveria. Precisamos de mais formação e treinamento em engenharia de software para solucionar esse problema

História da Engenharia de Software

O conceito de engenharia de software foi proposto pela primeira vez em 1968, em uma conferência realizada para discutir o que então se chamava crise do software (NAUR; RANDELL, 1969).

Ficou claro que as abordagens individuais ao desenvolvimento de programas não escalavam para sistemas de software grandes e complexos.

Os sistemas não eram confiáveis, custavam mais do que o previsto e eram entregues com atraso. Durante os anos 1970 e 1980, foi desenvolvida uma série de técnicas e métodos de engenharia de software, como a programação estruturada, a ocultação da informação (information hiding) e o desenvolvimento orientado a objetos.

Foram desenvolvidas ferramentas e notações que compõem a base da engenharia de software atual.

Desenvolvimento de Software Profissional

A engenharia de software se destina a apoiar o desenvolvimento de software profissional em vez de a programação individual.

Ela inclui técnicas que apoiam a especificação, o projeto e a evolução do software, aspectos geralmente irrelevantes para o desenvolvimento de software pessoal

Perguntas Frequentes

O que é software?

Programas de computador e documentação associada. Os produtos de software podem ser desenvolvidos para um determinado cliente ou para um mercado genérico.

Quais são os atributos do bom software?

O bom software deve proporcionar a funcionalidade e o desempenho necessários e deve ser manutenível.

O que é engenharia de software?

A engenharia de software é uma disciplina de engenharia que se preocupa com os aspectos da produção de software, desde sua concepção inicial até sua operação e manutenção.

Perguntas Frequentes

Quais são as atividades fundamentais da engenharia de software?

Especificação, desenvolvimento, validação e evolução do software.

Quais são os melhores métodos e técnicas de engenharia de software?

Ainda que todos os projetos de software devam ser gerenciados e desenvolvidos profissionalmente; técnicas diferentes são adequadas para tipos diferentes de sistemas.

Por exemplo, jogos devem ser sempre desenvolvidos usando uma série de protótipos enquanto sistemas de controle críticos em segurança requerem o desenvolvimento de uma especificação completa e analisável. Não há métodos ou técnicas que sejam bons para todos os casos.

Perguntas Frequentes

Quais diferenças a internet trouxe para a engenharia de software?

A internet não só levou ao desenvolvimento de sistemas massivos largamente distribuídos, baseados em serviços, como também deu base para a criação de uma indústria de aplicativos (ou "apps") para dispositivos móveis que mudou a economia de software

Se estiver escrevendo um programa para si próprio, que ninguém mais vai utilizar, não é necessário se preocupar em escrever manuais de programa, em documentar o projeto e assim por diante.

No entanto, se estiver desenvolvendo um software que outras pessoas utilizarão e no qual outros engenheiros farão alterações, então provavelmente será preciso fornecer informações adicionais sobre o programa, assim como seu código.

Tipos de Produtos

Existem dois tipos de produto

Produtos genéricos. São sistemas stand-alone produzidos por uma organização de desenvolvimento de software e vendidos no mercado para qualquer cliente que queira comprá-los.

Os exemplos desse tipo de produto incluem aplicativos para dispositivos móveis e software para PCs — como bancos de dados, processadores de texto, pacotes de desenho e ferramentas de gerenciamento de projetos. Esse tipo de software também inclui aplicações “verticais”, projetadas para um mercado específico, como sistemas de informação de bibliotecas, sistemas contábeis ou sistemas para manter registros odontológicos.

Software personalizado (ou feito sob medida). São sistemas encomendados e desenvolvidos para um determinado cliente.

Uma empresa de software projeta e implementa o software especialmente para o cliente.

Exemplos desse tipo de software incluem sistemas de controle para dispositivos eletrônicos, sistemas escritos para apoiar processos de negócios específicos e sistemas de controle de tráfego aéreo.

Tipos de Produtos

No entanto, a distinção entre esses tipos de produto está ficando cada vez mais confusa.

Mais e mais sistemas estão sendo criados tendo como base um produto genérico que depois é adaptado para atender às necessidades de um determinado cliente. Os sistemas ERP (Enterprise Resource Planning, em português, planejamento de recursos empresariais), como os da SAP e da Oracle, são os melhores exemplos dessa abordagem: um sistema grande e complexo é adaptado para uma empresa, incorporando informações sobre as regras e os processos de negócio, relatórios necessários e assim por diante

Atributos de Bom Software

Aceitabilidade

O software deve ser aceitável para o tipo de usuário para o qual é projetado. Isso significa que ele deve ser inteligível útil e compatível com os outros sistemas utilizados pelos usuários.

Dependabilidade e segurança da informação (security)

A dependabilidade do software induz uma gama de características. incluindo confiabilidade, segurança da informação (security) e segurança (safety). O software com dependabilidade não deve causar danos físicos ou econômicos em caso de falha do sistema.

Ele também deve ser protegido para que usuários maliciosos não consigam acessar ou danificar o sistema.

Eficiência

O software não deve desperdiçar recursos do sistema, como a memória e os ciclos de processador. Portanto, a eficiência inclui responsividade, tempo de processamento, utilização de recursos etc.

Atributos de Bom Software

Manutenibilidade

O software deve ser escrito de tal modo que possa evoluir e satisfazer as necessidades mutáveis dos clientes. Este é um atributo crítico, pois a modificação do software é um requisito inevitável de um ambiente empresarial mutável.

Engenharia de Software

Disciplina de engenharia. Os engenheiros fazem as coisas funcionarem. Eles aplicam teorias, métodos e ferramentas onde for apropriado.

No entanto, eles utilizam esses recursos seletivamente e sempre tentam descobrir soluções para os problemas, mesmo quando não há teorias e métodos aplicáveis.

Os engenheiros também reconhecem que devem trabalhar dentro de limites organizacionais e financeiros e que devem buscar soluções dentro desses limites.

Todos os aspectos da produção de software. A engenharia de software não se preocupa apenas com os processos técnicos do desenvolvimento de software. Ha também inclui atividades como o gerenciamento de projetos de software e o desenvolvimento de ferramentas, métodos e teorias que apoiam a criação de software.

4 atividades fundamentais processos de software

1. Especificação do software, etapa em que clientes e engenheiros definem o software que deve ser produzido e as restrições impostas à sua operação.
1. Desenvolvimento de software, etapa em que o software é projetado e programado.
1. Validação de software, etapa em que o programa é analisado para garantir que seja aquilo de que o cliente precisa.
4. Evolução do software, etapa de modificação para refletir a mudança de requisitos tanto do cliente quanto do mercado.

