

Fundamentos de Engenharia de Software

Josino Rodrigues (josinon@gmail.com)

Apresentação

- Quem sou eu
- Quem são vocês? Qual seu nível de conhecimento associado a disciplina e quais suas expectativas?

Objetivo da aula

- Depois desta aula você terá uma revisão sobre o que é a engenharia de software, os seus objetivos e conceitos básicos.

Leitura recomendada

- Capítulo 1 (Software and Software Engineering) do livro Software Engineering — A Practitioner's Approach (de Roger Pressman)
- Capítulo 1 (Introduction) do livro Software Engineering (de Ian Sommerville)

O que é a Engenharia de Software?

- Estudo ou aplicação de abordagens sistemáticas, econômicas e quantificáveis para o desenvolvimento, operação e manutenção de software de qualidade.
- Engenheiros de software devem adotar uma **abordagem sistemática e organizada** para seu trabalho e usar **ferramentas e técnicas/métodos apropriados** dependendo do problema a ser solucionado, das restrições de desenvolvimento e dos recursos disponíveis

O que é software?

- Programas de computador e documentação associada
- Produtos de software podem ser desenvolvidos para um cliente particular ou podem ser desenvolvidos para um mercado geral

Objetivos da Engenharia de Software

- Controle sobre o desenvolvimento de software dentro de **custos, prazos** e níveis de **qualidade** desejados
- Produtividade no desenvolvimento, operação e manutenção de software
- Qualidade versus Produtividade
- Permitir que profissionais tenham controle sobre o desenvolvimento de software dentro de custos, prazos e níveis de qualidade desejados

Características da Engenharia de Software

- A Engenharia de Software se refere a software (sistemas) desenvolvidos por grupos ao invés de indivíduos
- usa princípios de engenharia ao invés de arte, e
- inclui tanto aspectos técnicos quanto não técnicos

O que é um software de qualidade?

- O software que satisfaz os requisitos solicitados pelo usuário. Deve ser fácil de manter, ter boa performance, ser confiável e fácil de usar
- Alguns atributos de qualidade
 - Manutenibilidade
 - O software deve evoluir para atender os requisitos que mudam
 - Eficiência
 - O software não deve desperdiçar os recursos do sistema
 - Usabilidade
 - O software deve ser fácil de usar pelos usuários para os quais ele foi projetado

Qualidade de Software (um exemplo para o Varejo)

- Correto
 - A loja não pode deixar de cobrar por produtos comprados pelo consumidor
- Robusto e altamente disponível
 - A loja não pode parar de vender
- Eficiente
 - O consumidor não pode esperar
 - A empresa quer investir pouco em recursos computacionais (CPU, memória, rede)

Qualidade de Software (um exemplo para o Varejo)

- Amigável e fácil de usar
 - A empresa quer investir pouco em treinamento
- Altamente extensível e adaptável
 - A empresa tem sempre novos requisitos (para ontem!)
 - A empresa quer o software customizado do seu jeito (interface, teclado, idioma, moeda, etc.)
- Reusável
 - Várias empresas precisam usar partes de um mesmo sistema

Qualidade de Software

(um exemplo para o Varejo)

- Aberto, compatível, de fácil integração com outros sistemas
 - A empresa já tem controle de estoque, fidelização, etc.
- Portável e independente de plataforma (hw e sw)
 - A empresa opta por uma determinada plataforma
- Baixo custo de instalação e atualização
 - A empresa tem um grande número de PDVs

Produtividade

- Custo de desenvolvimento reduzido
 - A empresa consumidora quer investir pouco em software
 - A empresa produtora tem que oferecer “software barato”
- Tempo de desenvolvimento reduzido
 - Suporte rápido às necessidades do mercado

“Software Barato”

Nem tanto resultado de baixos custos de desenvolvimento, mas principalmente da distribuição dos custos entre vários clientes.

Reuso, extensibilidade e adaptabilidade são essenciais para viabilizar tal distribuição.

Importância da Engenharia de Software

- Qualidade de software e produtividade garantem:
 - Disponibilidade de serviços essenciais
 - Segurança de pessoas
 - Competitividade das empresas
 - Produtores
 - Consumidores

Mas, na realidade, temos a Crise de Software...

- 25% dos projetos são cancelados
- o tempo de desenvolvimento é bem maior do que o estimado
- 75% dos sistemas não funcionam como planejado
- a manutenção e reutilização são difíceis e custosas
- os problemas são proporcionais a complexidade dos sistemas

Causas da Crise de Software

- Essências
 - Complexidade dos sistemas
 - Dificuldade de formalização
- Acidentes
 - Má qualidade dos métodos, linguagens, ferramentas, processos, e modelos de ciclo de vida
 - Falta de qualificação técnica

Elementos e Atividades da Engenharia de Software

- Elementos

- Modelos do ciclo de vida do software
- Linguagens
- Métodos
- Ferramentas
- Processos

- Atividades

- Modelagem do negócio
- Elicitação de requisitos
- Análise e Projeto
- Implementação
- Testes
- Distribuição
- Planejamento
- Gerenciamento
- Gerência de Configuração e Mudanças
- Manutenção

Atividades e Artefatos da Engenharia de Software

- Atividades

- Modelagem do negócio
- Elicitação de requisitos
- Análise e Projeto
- Implementação
- Testes
- Distribuição
- Planejamento
- Gerenciamento
- Gerência de Configuração e Mudanças
- Manutenção

- Artefatos

- Plano de Negócios
- Plano de Projeto
- Plano de Riscos
- Documento de Requisitos
- Mapeamentos A&P
- Documento de Caso de Uso
- Documento de Arquitetura
- Classes
- Documento de Testes
- Documento de Validação
- Manual do Sistema

Leitura adicional

- Daniel M. Berry. [Myths and Realities in Software Development](#).
- W. Wayt Gibbs. [Software's chronic crisis](#). Scientific American, September 1994.
- Alan Joch. [How software doesn't work](#). Byte, December 1995.

E depois de tudo...

- Uma breve explicação sobre a condução da disciplina
- Formação dos grupos
- Definição de uma idéia que deverá ser implementada pelos grupos
- <http://www.ibm.com/smarterplanet/br/pt/>