Aula

0

APRESENTAÇÃO

Marcos Silva

Engenharia de Software

ENGENHARIA DE SOFTWARE

Apresentação

marcosxsilva@yahoo.com.br
http://geocities.yahoo.com.br/marcosxsilva/

Capítulo

1

Engenharia de Software e Gerenciamento de Projeto

As práticas de Engenharia de Software são a solução para resolver as questões de prazos, custos e qualidade na produção de software.

O software nos dias de hoje é onipresente. Praticamente todo equipamento eletrônico inclui alguma espécie de software. Alguns usados para auxiliar a produção de uma fábrica, de escolas, em UTIs ou ainda para simples diversão.

Definição

A especificação, o desenvolvimento, o gerenciamento e a evolução desde sistemas de software compõem a disciplina de Engenharia de Software.

Mesmo os sistemas de software mais simples possuem uma alta complexidade inerente, desta forma os princípios de engenharia devem ser usados no seu desenvolvimento.

De modo geral, a Engenharia de Software pode ser descrita como uma disciplina da engenharia, na qual os engenheiros usam métodos e teorias da ciência da computação e as aplicam de forma eficiente para resolver problemas difíceis.

Devido a grande diversidade de tipos de software, várias técnicas foram desenvolvidas. Não é possível cobrir todas as abordagens, no entanto esse curso abordará técnicas universalmente consagradas de desenvolvimento de software que cobrem a maior faixa dos tipos de software existentes, especialmente os sistemas de grande escala.

O curso não enfocará uma técnica em particular, porém os trabalhos serão desenvolvidos usando Programação Orientada a Objetos através de UML e Java,

preferencialmente. Alguns paralelos com Análise Estruturada serão feitos de modo que o aluno possa adaptar-se a outras realidades.

ICON KEY
Informações importantes
Teste seu conhecimento
Exercícios práticos
Revisão

É utilizada uma notação especial neste material para auxiliar os alunos a identificarem os pontos importantes, ou tarefas que permitirão o melhor aproveitamento do curso. Ao se deparar com os ícones apresentados ao lado, o aluno deve dedicar uma atenção especial ao seu conteúdo. Os mesmo indicam informações importantes, testes para avaliar o conhecimento do assunto, regras para exercícios práticos ou ainda itens teóricos que os alunos devem revisar.

Público Alvo

Esse curso foi moldado para atingir alunos de graduação de Ciência da Computação ou Engenharia de Computação, previsto para o último ano da grade curricular. Espera-se dos alunos conhecimentos básicos de programação, de preferência em Linguagens Orientadas a Objetos, assim como ter domínio sobre as principais estruturas de dados, como listas, filas, pilhas, etc.

É imprescindível que o aluno tenha familiaridade com sistemas modernos de computadores.

Sobre as atividades práticas

O curso tem um enfoque prático, visando a aplicação de conceitos teóricos básicos na resolução de problemas específicos. É necessário que os alunos respeitem o escopo das atividades e seus respectivos prazos de entrega.

O Curso

A primeira parte do curso abordará os conceitos introdutórios da disciplina, tais como a importância da engenharia de software, as características de um software, o que é a crise do software e os mitos e paradigmas da engenharia de software. Nesta primeira parte ainda serão descritos os conceitos de engenharia de software e os princípios de gerenciamento de projetos.

Cronograma

O cronograma de atividades, composto de aulas e atividades práticas é planejado para trinta semanas, dividido em dois semestres de quatro créditos cada. Esse cronograma encontra-se em documento separado.

A segunda etapa consiste no estudo das técnicas de levantamento de requisitos e seu gerenciamento.

Na terceira etapa, serão estadados os tópicos relacionados ao design dos sistemas de software e as boas práticas de codificação.

Numa quarta etapa, o foco será nos processo de verificação e validação dos sistemas de software. Nesta fase, também será analisado o processo de testes de software.

Alguns assuntos, como gerenciamento de projetos, gerenciamento da garantia de qualidade, melhoria de processo e os processos de estimativas, serão vistos em paralelo com as etapas já mencionadas. As atividades descritas nesses assuntos formam um suporte importantíssimo no desenvolvimento de produtos de software.

Finalmente, serão abordados os processos relacionados à evolução e manutenção de sistemas de software, incluindo detalhes sobre a gerência de configuração. Em particular, alguns pontos sobre gerenciamento de configuração serão introduzidos ainda na segunda fase do curso.

Conteúdo

O curso será dividido nos seguintes tópicos.

Parte I

1. Introdução

- O que é Engenharia de Software
- Importância
- Crise do software
- Mitos

- Paradigmas
- Responsabilidade Ética e Profissional
- FAQs

2. Engenharia de Sistemas
    - Sistemas emergentes
    - Modelagem de Sistemas
    - Processo de engenharia de sistemas
    - Aquisição de sistemas
3. Processos de Software
    - Modelos
    - Processos interativos
    - Especificação de software
    - Design e implementação de software
    - Validação de software
    - Evolução e manutenção
    - Suporte a processos automatizados
    - Exemplos: PSP, SPICE, RUP
4. Gerenciamento de Projetos
    - Introdução a estrutura de gerenciamento de projetos
    - Planejamento de projetos e seus processos
    - Áreas de Conhecimento em Gerência de Projetos (PMBOK)
        - Integração
        - Escopo
        - Prazo
        - Custo
        - Qualidade
        - Recursos Humanos
        - Comunicação
        - Riscos

- Contratos e Aquisição

Parte II

5. Requisitos de Sistema e de Software
   - Requisitos funcionais/não-funcionais
   - Requisitos de usuário/sistema
   - Documento de requisitos
6. Processo de Engenharia de Requisitos
   - Estudos de viabilidade
   - Elicitação e análise
   - Validação dos requisitos
   - Gerenciamento de requisitos
   - Especificação de requisitos de software
7. Modelos de Sistemas
   - Modelos contextuais
   - Modelos comportamentais
   - Modelos de dados
   - Modelos de Objetos
   - CASEs
8. Prototipação
   - Prototipagem no processo de software
   - Técnicas de prototipagem
   - Design de interfaces

Parte III

9. Design de Arquitetura
   - Sistemas estruturados
   - Modelos de controle

- Decomposição modular
- Arquiteturas de domínio específico
- Sistemas de arquiteturas distribuídas
  - Arquiteturas multiprocessadas
  - Arquiteturas cliente-servidor
  - Objetos distrubuídos

10. Design Orientado a Objetos
    - Classes e objetos
    - Processo de software orientado a objetos
    - UML
    - Componentes e reuso
    - Design Patterns
11. Design de Interface com o usuário
    - Princípios
    - Interação com o usuário
    - Apresentação da informação
    - Suporte e documentação ao usuário
    - Heurísticas de avaliação
12. Codificação
    - Diretrizes genéricas
    - Java
    - Boas práticas de codificação

Parte IV

13. Verificação e Validação
    - Planejamento de verificação e validação
    - Inspeções e revisões de produtos e software
    - Análise estática

14. Testes
    - Testes de defeitos
    - Testes unitários
    - Testes de integração
    - Testes de sistema
    - Testes orientados a objeto
    - Testes de regressão
    - Noções de TTCN

Parte V

15. Fundamentos Gerenciais
    - Plano de Desenvolvimento de Software
    - Plano da Garantia da Qualidade
    - Métricas de Software
    - Acompanhamento de projetos de software
16. Melhorias nos Processos de Software
    - Qualidade de Produtos de software
    - Qualidade de Processo de desenvolvimento de software
    - Processos de medidas e verificações
    - SEI/CMM

Parte VI

17. Sistemas legados
    - Estruturas
    - Design
    - Avaliação
18. Mudanças no Software
    - Programa de evoluções dinâmicas

- Manutenção de software
- Gerenciando os pedidos de mudanças
- Rastreabilidade

19. Gerenciamento de Configurações
    - Planejamento da gerência de configuração
    - Versionamento
    - Gerência de *releases*
    - "*System building*"
    - Ferramenta CASE para Gerenciamento de Configuração

Avaliações

A avaliação dos alunos será composta de quatro itens. Os prazos de entrega serão definidos no cronograma em arquivo separado.

1. Projeto

As turmas serão dividas em grupos de dez alunos. Cada grupo desenvolverá um projeto de software, valendo 50% da nota. O projeto deverá apresentar os seguintes artefatos de software:

T1: Documento de Escopo

T2: Documento de Casos de Uso

T3: Documento de Especificação de Requisitos de Software, com matriz de rastreabilidade para os Casos de Uso

T4: Documento de Design e Planos de Testes Unitários

T5: Código Inspecionado e matrizes de Testes Unitários

T6: Planos de testes de Integração e de Sistema

T7: Matriz de Resultados de Testes

T8: Matrizes de Rastreabilidade

T9: *Pós Mortem*

O Modelo de Processo de Desenvolvimento de Software será definido com o escopo do projeto e estará disponível aos alunos, assim como todos os *templates* necessários.

2. Planos de Projeto

Cada projeto é regido por planos. Cada equipe deverá escrever seus respectivos planos, totalizando 25% da nota.

P1: Plano de Desenvolvimento de Software

P2: Plano de Garantia da Qualidade de Software

P3: Plano de Gerência de Configuração

P4: Matrizes de Acompanhamento de Projeto

3. Avaliação Individual

Ao longo do curso serão aplicadas provas teóricas, as quais os alunos resolverão individualmente. Será aplicada uma prova teórica por bimestre que corresponderá a 25% da nota final do aluno. As datas das mesmas serão agendadas no cronograma em arquivo anexo.

Cada prova será composta de quatro questões objetivas e abordarão os tópicos descritos no conteúdo do curso e nas atividades práticas desenvolvidas no período. Será permitido o uso do livro texto, notas de aula e produtos de software produzidos no período como material de consulta e apoio nessas avaliações.

Para permitir uma preparação para essas provas teóricas, cada tópico estudado conterá uma série de questões ou exercícios a serem resolvidas pelos alunos. Não haverá necessidade de entrega dessas questões. O docente irá comenta-las e apontar qual o raciocínio a ser usado na solução. Em caso de dúvidas, o docente auxiliará os alunos na solução.

Requisitos de Software/Hardware

Para o desenvolvimento dos projetos serão necessários:

1. Computadores com arquitetura Intel x86, com recursos de hardware compatíveis com as atividades.
2. Ambiente de desenvolvimento para Windows NT/2000.
3. Ferramentas de microinformática, tais como editores de texto, planilhas e apresentações.
4. Projetor multimedia
5. CASE para Gerência de Requisitos. Preferencialmente alguma solução via WEB. Uma boa opção é o DOORS.

6. Ferramenta CASE para UML. Uma boa opção freeware é o ArgoUML.

7. CASE para Gerência de Configurações. Uma boa opção é o CVS e seus front-ends.

8. Ferramenta para Planejamento e Acompanhamento de projetos. Recomenda-se o uso do Microsoft Project.

9. Sistema de Gerenciamento de Banco de Dados. Preferencialmente alguma opção freeware, como o MySQL.

10. IDE para desenvolvimento em Java. (Forte, da Sun MicroSystems; Visual Age for Java, da IBM).

11. Servidor, no caso de opção pela arquitetura cliente-servidor, ou arquitetura WEB. Neste caso utilizar soluções abertas, como Apache para servidor WEB e o Tomcat, do projeto Jakarta como servidor de aplicação.

Referências

Ian Sommerville; Software Engineering; 6th Edition, **Addison-Wesley (2001)**. ISBN 0 201 39815 X

Roger Pressman; Software Engineering: A Practitioner's Approach; 5th Edition; **McGraw Hill (1997)**