

"Disciplinas da Formação em Engenharia de Software"

Engenharia de Software

Engenharia de Requisitos

Desenvolvimento Distribuído

Teste e Qualidade de Software

Gerência de Projetos

Engenharia de Software - 30 hs

Objetivo:

Fornecer uma visão sistêmica da Engenharia de Software, visando desenvolver ou aperfeiçoar a capacidade do aluno de: atuar em projetos de engenharia de software, avaliar tecnologias (modelos, métodos, técnicas e ferramentas) já existentes e investigar novas tecnologias para apoio às atividades de engenharia de software.

Ementa:

Engenharia de Software: conceito, histórico e tendências. Engenharia e Gerência de Requisitos. Projeto de Software: princípios, arquitetura, padrões e *frameworks*. Construção de Software: princípios, documentação, gerência, reuso. Manutenção de Software: tipos, processo, questões gerenciais e técnicas, engenharia reversa e reengenharia. Gerência de Configuração: atividades e conceitos associados. Processos de Software: tipos, definição, medição e avaliação. Qualidade de Software: conceito, o que controlar, quando controlar, como controlar, quem deve controlar. Gerência da Engenharia de Software: definição do escopo, planejamento, monitoração, avaliação. Ferramentas e Ambientes de Desenvolvimento de Software.

Bibliografia básica:

1) Guide to the Software Engineering Body of Knowledge, IEEE Computer Society, 2004.

<http://www.swebok.org/>.

2) Sommerville, I. - Software Engineering, 8th edition, Addison Wesley, 2007.

3) Pressman, R. S. - Software Engineering: A Practitioner's Approach. 7th Edition, Mc Graw Hill, 2009. .

Bibliografia complementar:

4) Capability Maturity Model Integration (CMMI) Version 1.1 - Staged Representation, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, 2002.

5) Pfleeger, S. L. - Software Engineering : Theory and Practice, 1st edition, Prentice Hall, 1998

Engenharia de Requisitos – 30 horas

Objetivo:

Fornecer ao aluno uma visão sobre o processo de engenharia de requisitos, os conceitos fundamentais e importância da engenharia de requisitos no processo de desenvolvimento de software. Apresentar as atividades envolvidas evidenciando a importância de cada uma delas e os diferentes produtos gerados, bem como o relacionamento entre elas. Desenvolver senso crítico do aluno mostrando a necessidade do gerenciamento de requisitos para apoiar cada atividade do ciclo de vida.

Ementa:

Requisitos de software; Engenharia de requisitos: conceitos e definições; Engenharia de requisitos em diferentes ciclos de vida; O processo de engenharia de requisitos: etapas e produtos gerados. Engenharia de Requisitos; Gerenciamento de Requisitos; Evolução de Requisitos; Rastreabilidade de Requisitos.

Bibliografia básica:

- 1) WIEGERS, K. E., More About Software Requirements: Thorny Issues and Practical Advice, Microsoft Press, 2006.
- 2) ZIELCZYNSKI, P., Requirements Management Using IBM Rational RequisitePro”, IBM Press, 2007.

Bibliografia complementar:

- 3) BOOCH, G.; RUMBAUGH, J.; JACOBSON, I., UML: Guia do Usuário”, Rio de Janeiro: Elsevier, 2005.
- 4) PRESSMAN, R., Engenharia de Software”, 6.ed. - São Paulo: McGraw-Hill, 2006.
- 5) SOMMERVILLE, I., Software Engineering”, 8th. ed. Addison Wesley, 2007.
- 6) MYLOPOULOS, J., CHUNG, L., NIXON, B., Representing and Using Nonfunctional Requirements: A Process-Oriented Approach, IEEE Transactions on Software Engineering, Vol. 18, No. 6, June, 1992.

Desenvolvimento Distribuído – 30 horas**Objetivos:**

Fornecer ao aluno condições para a aplicação dos conceitos de colaboração nas diferentes etapas do processo de desenvolvimento distribuído de software. Identificar as facilidades e os desafios existentes ao utilizarmos ferramentas colaborativas para apoiar o desenvolvimento distribuído de software. Identificar os requisitos básicos para a construção de sistemas para o suporte ao desenvolvimento distribuído de software.

Ementa:

A área de CSCW e as ferramentas para apoiar o desenvolvimento distribuído de software. Conceitos e características necessárias para ferramentas colaborativas. Modelagem de processos e ferramentas para apoiar as atividades de desenvolvimento distribuído de software. Ciclo de vida da engenharia de software e as etapas nas quais a colaboração necessita de apoio computacional.

Bibliografia básica:

- 1) COLEMAN, David. “Groupware: Collaborative Strategies for Corporate LANs and Intranets”. Prentice Hall, 1997.
- 2) SANGWAN, R., BASS, M., MULLICK, Neel, PAULISH, D. J. “Global Software Development Handbook”, Auerbach Series on Applied Software Engineering Series, 2006.
- 3) PRIKLADNICKI, R., AUDY, J. L. N. “Desenvolvimento Distribuído de Software”, Ed. Campus, 2007.

Bibliografia complementar:

- 4) CRUZ, T. “Uso e desuso de sistemas de Workflow: Porque as organizações não conseguem obter retorno, nem sucesso, com investimentos em projetos de Workflow”. Rio de Janeiro: E-Papers Serviços Editoriais, 2006.
- 5) KHOSHAFIAN, S.; BUCKIEWICZ, M. “Introduction to Groupware, Workflow, and Workgroup Computing”, John Wiley & Sons, Inc.. 1995.
- 6) CHAFFEY, D. “Groupware, workflow and intranets: reengineering the enterprise with collaborative software”. Digital Press, 1998.
- 7) SOMMERVILLE, I. “Engenharia de Software”. 8ª. ed. – São Paulo: Pearson Addison-Wesley, 2007.

8) PRESSMAN, R. S. "Engenharia de Software". 6. ed. – São Paulo: McGraw-Hill, 2006.

9) CRUZ, T. "Workflow: a tecnologia que vai revolucionar processos". 2. ed. – São Paulo: Atlas, 2000.

Teste e Qualidade de Software – 30 horas

Objetivo: Apresentar as atividades de verificação, validação e teste de software no ciclo de vida da Engenharia de Software. Compreender a importância das atividades de teste, bem como as técnicas e as estratégias relacionadas ao processo. Discutir os diferentes tipos de teste de software e as suas utilizações. Apresentar a importância do gerenciamento da qualidade no processo de desenvolvimento de software.

Ementa: Teste no processo de desenvolvimento de software; Partições de equivalência e dados de teste; Verificação e validação; Projeto de casos de teste; Teste de validação; Planejamento do teste; Casos de teste; Teste e Processos de software; Testes automatizados; A qualidade do produto e do processo de desenvolvimento de software; Planejamento e Controle de qualidade; Medição e métricas de software; Normas de qualidade; CMMI, MPS-BR, ISO.

Bibliografia básica:

1) DELAMARO, M. E., MALDONADO, J. C., JINO, M., Introdução ao Teste de Software, Rio de Janeiro: Elsevier, 2007.

2) PATTON, R., Software Testing, 2nd. ed., Sams Publishing, 2006.

3) BARTIÉ, A., Garantia da qualidade de software. 1.ed. Rio de Janeiro: Elsevier, 2002.

Bibliografia complementar:

4) FAGAN, M. E., Advances in software inspections, IEEE trans. On software engineering, SE-12(7), 744-51, 1986.

5) MILLS, H. D. DYER, M. et al., Cleanroom software engineering, IEEE software, 4(5), 19-25, 1987.

6) PRESSMAN, R., Engenharia de Software, 6.ed. - São Paulo: McGraw-Hill, 2006.

7) SOMMERVILLE, I., Software Engineering, 8th. ed., Addison Wesley, 2007.

8) SWEBOK, Guide to the Software Engineering Body of Knowledge, IEEE Computer Society, <http://www.swebok.org/>.

9) KOSCIANSKI, A., SOARES, M. S., Qualidade de Software: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software, São Paulo: Novatec Editora, 2006.

Gerência de Projetos – 30 horas

Objetivo: Apresentar os conceitos fundamentais para a gerência de projetos de software, bem como as etapas de projeto com o foco em sistemas de informação. Capacitar o aluno para o gerenciamento de projetos voltado para os conceitos abordados na metodologia do Project Management Institute (PMI).

Ementa: Objetivos da gerência de projetos; Definição do escopo; Planejamento do projeto; Estimativas; Gerenciamento das atividades; Avaliação e revisão; Gerenciamento de Riscos; Encerramento; Desenvolvimento e implementação do programa de medidas.

Bibliografia básica:

1) VAZQUEZ, C. E., SIMÕES, G. S., ALBERT, R. M. Análise de pontos de função: medição, estimativas e gerenciamento de projetos de software, 1. ed. – São Paulo: Érica, 2003.

2) Project Management Institute. A guide to the project management body of knowledge. USA, 2004.

Bibliografia complementar:

- 3) MARTINS, J. C. C., Gerenciando Projetos de Desenvolvimento de Software com PMI, RUP e UML, 4. Ed., Rio de Janeiro,: Brasport, 2007.
- 4) PRESSMAN, R., Engenharia de Software, 6.ed. - São Paulo: McGraw-Hill, 2006.
- 5) SOMMERVILLE, I., Software Engineering, 8th. ed. Addison Wesley, 2007.
- 6) SWEBOK, Guide to the Software Engineering Body of Knowledge, IEEE Computer Society, <http://www.swebok.org/>, 2004.