



UNIVERSIDADE ESTADUAL DE SANTA CRUZ-UESC
PRÓ-REITORIA DE GRADUAÇÃO – PROGRAD
DEPARTAMENTO DE CIÊNCIAS EXATAS - DCET
COLEGIADO DE CIÊNCIA DA COMPUTAÇÃO – COLCIC

PROGRAMA DE DISCIPLINA

CÓDIGO	DISCIPLINA	PRÉ-REQUISITO(S)
CET 228	ENGENHARIA DE SOFTWARE	ANÁLISE DOS SISTEMAS DE INFORMAÇÃO

C/HORÁRIA	CRÉDITOS	PROFESSOR (A)
T 45	3	
P 30	1	
TOTAL 75	4	

EMENTA

Software e Engenharia de Software- Características, Paradigmas e Visão Geral; Sistemas de Software, Hardware, Base de Dados e Sistemas de Informação; Planejamento do Projeto do Software; Fundamentos (Tarefas de Análise de Requisitos e Áreas Problemáticas) e Métodos (Orientados a Fluxo de dados, Estrutura de Dados, Tempo Real e orientação a objetos); Garantia de Qualidade de Software; Revisões , Técnicas Formais, Métricas; ; Técnicas e Estratégias de Teste; Técnicas Funcional, Estrutural e Baseadas em Erros; teste de Unidade, de Integração e de sistema; manutenção e gerenciamento de Configurações; Definição, Características, Tarefas e tipos

OBJETIVOS

Desenvolver habilidades para o gerenciamento e execução de projetos de desenvolvimento de software, utilizando técnicas de modelagem de dados e sistemas baseados em computadores, aplicando procedimentos de engenharia e administração de empresas visando a garantia da qualidade dos produtos de software.

METODOLOGIA

Aprendizado Baseado em Problemas, estruturando equipes de desenvolvimento de software que funcionam como microempresas. Inclui uma fase preparatória que combina aulas expositivas e seminários por equipes, uma fase de aquecimento com dinâmicas de grupo, e uma fase de execução dos projetos.

AVALIAÇÃO

Avaliação de seminários, prova parcial e acompanhamento de projetos.
Apresentação do projeto.

CONTEÚDO PROGRAMÁTICO

1. Fundamentos
 - 1.1 Evolução do Software
2. Paradigmas
 - 2.1 Modelo cascata
 - 2.2 Prototipação
 - 2.3 Modelo espiral
 - 2.4 Técnicas de 4a geração
3. Sistemas de informação
4. Sistemas de tempo real
 - 4.1 Redes de petri
5. Gerência de projetos
6. Métricas
 - 6.1 Pontos de função
7. Estimativas
8. Planejamento
 - 8.1 Análise de riscos
 - 8.2 Cronogramas
 - 8.3 Tarefas
9. Requisitos
 - 9.1 Alternativas
 - 9.2 Viabilidade
 - 9.3 Arquitetura
 - 9.4 Comunicação
 - 9.5 Prototipação
 - 9.6 Especificação
10. Análise
 - 10.1 DFD – Diagrama de Fluxo de Dados
 - 10.2 Dicionário de dados
 - 10.3 Diagrama de transição de estados
11. Projeto
 - 11.1 Conceitos: abstração, refinamento, modularidade, etc.
12. Análise OO
 - 12.1 Identificação de objetos
 - 12.2 Atributos, operações e mensagens
 - 12.3 Diagramas
13. UML
14. Implementação
15. Manutenção
16. Teste de software
17. Gerenciamento de configurações
18. Qualidade de software: O modelo CMMI

REFERÊNCIA BIBLIOGRÁFICA

A M. Davis. **Software Requirements: Objects, Functions and States.** Prentice-Hall, Upper Saddle River-NJ., 1993.

B. Shneiderman. **Designing the user interface: strategies for effective human-computer interaction.** Addison-wesley, 2^a edição, 1992.

D.A Taylor. **Object-Oriented Technology: a Manager's Guide.** Addison-Wesley, Reading MA, 1990

G. Booch , I. Jacobson, J. Rumbaugh. **Unified Modeling Language User Guide.** Addison-Wesley, Reading –MA, 1999.

G. Booch. **Object-Oriented Analysis and Design with Applications.** 2nd. Ed. Benjamin/Cummings, Redwood City-CA, 1994.

G. Booch. **Object solutions: Managing the Object-Oriented Project.** Addison-Wesley, Reading-MA, 1996. F. P. Brooks, Jr. **The Mythical Man-Month.** Addison-Wesley, reading-MA, 1996.

G. Schneider, J.P. Winter. **Applying Use Cases: A Practical Guide.** Addison Wesley, Reading-MA, 1998.

J. Rumbaugh, I. Jacobson, G. Booch. **Unified Modeling Language Reference Manual.** Addison-Wesley, Reading –MA, 1999.

L.L. Constantine, L.A D. Lockwood. **Software for Use: Practical Guide to the Models and Methods of Usage-Centered Design.** Addison-weley, Reading-MA, 1999.

M. Blaha, W. Premerlani. **Object Oriented Modeling and Design for Database Applications.** Prentice-Hall, Englewood Cliffs, NJ, 1998.

M.C. Paulk, C.V. Weber, B. curtis, M.B. Chrissis. **The Capability Maturity Model: Guidelines for Improving Software Process.** Addison-Wesley, Reading-MA, 1995.

R. Pressman. **Engenharia de Software.** 3^a ed. Makron Books do Brasil, São Paulo, 1995.

S. Robertson, J. Robertson. **Mastering the Requirement Process.** Addison-Wesley, Harlow-england, 1999.

S.T. Fiorini, A V. Staa, R.M. Baptista. **Engenharia de Software com CMM.** Brasport, 1998.

W. S. Humphrey. **Introduction to the Team Software Process.** Addison-Wesley, Reading-MA, 1999.

W. P. Paula Filho, **Engenharia de Software**, LTC, 2003.