

MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA
DIRETORIA DE ENSINO (DIREN)
DEPARTAMENTO DE ENSINO SUPERIOR (DEPES)
DEPARTAMENTO DE INFORMÁTICA (DEPIN)
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO (BCC)

DEPARTAMENTO
DEPIN - Departamento Acadêmico de Informática

PLANO DE CURSO DA DISCIPLINA
TESTE DE SOFTWARE

CÓDIGO
GCC1929

PERÍODO
Opt

ANO
2012

SEMESTRE
2

PRÉ-REQUISITOS
GCC 1311 Programação Orientada a Objetos
GCC 1312 Engenharia de Requisitos

CRÉDITOS
4

AULAS/SEMANA		
TEÓRICA	PRÁTICA	ESTÁGIO
4	0	0

TOTAL DE AULAS NO SEMESTRE
72

EMENTA

Conceitos de qualidade, testes, verificação e validação de software. Processo de Teste de Software. Fases de Testes. Projeto e técnicas de geração de caso de testes. Métricas e adequação dos testes. Teste de requisitos não funcionais. Documentação dos Testes. Automação e ferramentas de testes.

BIBLIOGRAFIA

Bibliografia básica

1. FREEMAN, Steve; PRYCE, Nat. Growing object-oriented software, guide by tests. Boston: Addison-Wesley. 358 p., il. ISBN 9780321503626.
2. BECK, Kent. TDD desenvolvimento guiado por testes. Porto Alegre: Bookman. 240p., il. ISBN 9788577807246.
3. BARTIÉ, Alexandre. Garantia da qualidade de software. Rio de Janeiro: Elsevier: Campus. 291 p., il. ISBN 9788535211245.

Bibliografia complementar

1. PRESSMAN, Roger S., Engenharia de software: uma abordagem profissional. Porto Alegre, RS: AMGH Ed., 780 p., il. Bibliografia: p. [751]-771. ISBN 9788563308337.
2. PRESSMAN, Roger S.; LOWE, David Brian., Engenharia web. Rio de Janeiro: LTC. 416 p. ISBN 9788521616962.
3. SOMMERVILLE, Ian. Engenharia de software. São Paulo: Pearson. 529p., il. ISBN 9788579361081.
4. PAULA FILHO, Wilson de Pádua. Engenharia de software: fundamentos, métodos e padrões. Rio de Janeiro: LTC Ed, 1248p, il. Bibliografia p. [1235]-1244. ISBN 9788521616504.
5. WAZLAWICK, Raul Sidnei. Engenharia de software: conceitos e práticas. Rio de Janeiro: Elsevier. p.[355]-338. ISBN 9788535260847.

OBJETIVO GERAL

Entender porque é importante testar software. Conhecer o processo aplicado para testar software. Entender as diversas técnicas de geração de casos de teste e quando cada é aplicável. Conhecer as principais ferramentas de apoio ao teste de software.

METODOLOGIA

- Aulas expositivas, eventualmente contando com recursos audiovisuais.
- Resolução de exercícios de fixação e propostos.

CRITÉRIO DE AVALIAÇÃO

A avaliação semestral envolve duas provas escritas (P1 e P2). As datas das provas são agendadas conforme o calendário da Instituição. A média parcial (MP) será calculada pelo cômputo da média aritmética simples entre a nota P1 e P2:

$$MP = (P1 + P2) / 2$$

O aluno que faltar a uma das duas provas terá direito a uma avaliação alternativa, denominada segunda chamada, versando sobre todos os tópicos abordados no curso, e cuja data também é agendada entre docente e discentes. A nota obtida nessa 2ª chamada substituirá a da avaliação P1 ou P2 onde o aluno não esteve presente. Caso ele falte às duas avaliações, terá atribuído o grau ZERO em uma delas.

Segundo o regimento do CEFET-RJ, caso o aluno obtenha média parcial inferior a 3,0 (três e zero) estará reprovado diretamente. Graus MP maiores ou iguais a 7,0 (sete e zero) aprovam diretamente o aluno. Em situações onde o aluno tenha grau MP entre 3,0 inclusive e 7,0 exclusive, terá direito a uma prova final (PF), que, juntamente com a média parcial gerará uma nova média, denominada média final (MF). Essa média é calculada da seguinte forma:

$$MF = (MP + PF) / 2$$

Para ser aprovado, o aluno deve alcançar uma MF maior ou igual a 5,0 (cinco e zero). Caso contrário, estará reprovado, devendo repetir a componente curricular.

PROGRAMA

1. Conceitos de qualidade, testes, verificação e validação de software
2. Processo de Teste de Software
3. Fases de Testes
 - 3.1. Unitário
 - 3.2. Integração
 - 3.3. Funcional
 - 3.4. Sistema
4. Projeto e técnicas de geração de caso de testes
 - 4.1. Baseados em especificação
 - 4.2. Baseados no código-fonte
 - 4.3. Baseados em modelos
 - 4.4. Orientado a objetos

5. Métricas e adequação dos testes
6. Teste de requisitos não funcionais
 - 6.1. Teste de desempenho, carga e estresse
 - 6.2. Teste de Usabilidade
 - 6.3. Teste de Segurança
7. Documentação dos Testes
8. Automação e ferramentas de testes