



UNIVERSIDADE PRESBITERIANA MACKENZIE



Unidade Universitária: FACULDADE DE COMPUTAÇÃO E INFORMÁTICA		
Curso: Ciência da Computação Sistemas de Informação Tecnologia em Análise e Desenvolvimento de Sistemas		
Disciplina: ENGENHARIA DE SOFTWARE II		Código da Disciplina: ENEC00147
Professor(es): CALEBE DE PAULA BIANCHINI KASSYA CHRISTINA R DE ANDRADE MARIA INES L BROSSO PIOLTINE	DRT: 1130664 1078681 1129146	Etapa: 04
Carga horária: 34 Teóricas, 34 Práticas, 0 EaD		Semestre Letivo: 2ºSEM/2016
<p>Ementa:</p> <p>Fundamentação de Qualidade de Software: Qualidade do Processo e Qualidade do Produto, Normas e Modelos. Fundamentação dos conceitos e processo de gerência de configuração e controle de versão. Domínio dos conceitos do processo e técnicas de teste de software. Gerência, Análise, Projeto, implementação e execução de testes.</p>		
Objetivos:		
<p><i>Fatos e Conceitos</i></p> <ul style="list-style-type: none"> - Aprender fundamentos teóricos gerais necessários à qualidade de software e do produto de software - Conhecer e compreender os processos fundamentais, de apoio e organizacionais do Ciclo de vida de software - Estudar e reconhecer as normas e modelos de maturidade para processos de software 	<p><i>Procedimentos e Habilidades</i></p> <ul style="list-style-type: none"> - Adquirir capacidade para desenvolver produtos de software com qualidade - Adquirir capacidade para praticar a garantia de qualidade - Adquirir capacidade para desenvolver programas com qualidade 	<p><i>Atitudes, Normas e Valores</i></p> <ul style="list-style-type: none"> - Ser capaz de identificar as tarefas inerentes a cada fase do processo de software - Ser capaz de aplicar técnicas de validação e verificação no desenvolvimento de software - Ser capaz de avaliar e compreender normas e modelos de maturidade - Ter consciência da necessidade de busca da qualidade de software em todas as fases no processo de desenvolvimento de software - Valorizar trabalho em equipe para desenvolvimento de tarefas complexas
<p>Conteúdo Programático:</p> <p>0. Apresentação do Plano de Ensino e dos Pré-Requisitos</p> <p>1. Qualidade de Software e Programas de Melhoria da Qualidade de Software. Certificação de equipes de desenvolvimento de sistemas.</p>		

2. Modelo Processo de Software e Documentação, Métricas de Software
3. Gerenciamento da Configuração de software e Controle de versões
4. Verificação e Validação. Técnicas e estratégias de teste de software
5. Certificação de equipes de desenvolvimento de sistemas.

Metodologia:

- Aulas expositivas - Listas de exercícios comentadas - Desenvolvimento de Trabalho em Grupo
- Utilização do ambiente Mackenzie Virtual

Critério de Avaliação:

A composição da média intermediária (MI) do aluno é feita através da seguinte fórmula:

$$NI1: 0,20 \cdot P1 + 0,10 \cdot LAB1 + 0,10 \cdot LAB2 + 0,10 \cdot LAB3$$

$$NI2: 0,20 \cdot P2 + 0,10 \cdot LAB1 + 0,10 \cdot LAB2 + 0,10 \cdot LAB3$$

$$PARC: 0,05 \cdot (\text{Avaliação diagnóstica}) + 0,05 \cdot (\text{Participação das Atividades}).$$

$$MI = (NI1 + NI2)/2 + Parc$$

MI - Média Intermediária

Sendo que:

- P1 (A) – Prova Parcial 1 (prova individual, sem consulta) (20%).
- LAB1 (B)– Atividade de laboratório (10%)
- LAB2 (C)– Atividade de laboratório (10%)
- LAB3 (D)- Atividade de laboratório (10%)
- P2 (F) – Prova Parcial 2 (prova individual, sem consulta) (20%).
- LAB4 (G)– Atividade de laboratório (10%)
- LAB5 (H)– Atividade de laboratório (10%)
- LAB6 (I)- Atividade de laboratório (10%)
- PARC: Avaliação diagnóstica (0,5) e Participação das Atividades desenvolvidas em aulas teóricas e laboratório (0,5). Participação: 0 a 1 ponto na média.

Prova Substitutiva substitui a menor nota da Média Intermediária. A prova substitutiva da disciplina será realizada nas últimas semanas de aula.

A composição da média Final (MF) do aluno é feita através da seguinte formula:

- Se a MI \geq 7,5 então MF = MI Senão MF = (MI + PF)/2

- PF – Prova Final: Prova individual, sem consulta, sobre a totalidade do conteúdo do semestre – 50%

Bibliografia Básica:

- CRAIG, Rick D.; JASKIEL, Stefan P. Systematic software testing. 6th printing Boston: Artech House, 2006.
- PEZZÈ, M.; YOUNG, M. Teste e análise de software: processo, princípios e técnicas. São Paulo: Bookman, 2008.
- SOMMERVILLE, I. Software engineering. 9ª ed. Harlow: Addison-Wesley, 2011.

Bibliografia Complementar:

- BECK, K. TDD Desenvolvimento Guiado por Testes. Porto Alegre: Grupo A, 2010.
- BURNSTEIN, I. Practical software testing: a process-oriented approach. 3ª printing New York:

Springer, 2003.

- COCKBURN, A. Agile software development: the cooperative game. 2ª ed. Harlow: Addison Wesley, 2007.

- SOARES, M. S.; KOSCIANSKI, A. Qualidade de Software. Editora: Novatec, 2006.

- WAZLAWICK, R. Engenharia de Software: Conceitos e Práticas. 1ª. Edição. Rio de Janeiro: Elsevier- Campus, 2013.