

Programa Analítico de Disciplina

INF 323 - Engenharia de Software II

Departamento de Informática - Centro de Ciências Exatas e Tecnológicas

Catálogo: 2019

Número de créditos: 4
Carga horária semestral: 60h
Carga horária semanal teórica: 4h
Carga horária semanal prática: 0h
Semestres: II

Objetivos

Ao final desta disciplina o estudante deverá ser capaz de: Aplicar na prática, conceitos de Engenharia de Software, suas áreas, seus problemas, e seu futuro; Definir e gerenciar processos baseados nos mais diversos Modelos de Processos de Desenvolvimento de sistemas; Utilizar conceitos e artefatos de Métodos Ágeis na customização de processos; Adaptar processos aos níveis de Modelos de maturidade organizacional CMMI e mpsBR para comprovar a capacidade de uma equipe em obter qualidade de software; Realizar Análise e Projeto de Sistemas com método OO; Tomar decisões a partir do entendimento de um contexto para planejamento de projetos e avaliação de riscos de projetos; Aplicar Métricas utilizadas na precificação de software, dimensionamento de equipes e estabelecimento de cronogramas.

Ementa

O contexto para Engenharia de Software. Processos de desenvolvimento de software. Análise e especificação de requisitos de software. Modelagem estática de sistemas. Modelagem dinâmica de sistemas. Modelagem física. Aspectos de implementação. Introdução à gerência de projetos de software.

Pré e co-requisitos

INF 221

Oferecimentos obrigatórios

Curso	Período
Ciência da Computação	6

Oferecimentos optativos

Não definidos

INF 323 - Engenharia de Software II

Conteúdo					
Unidade	T	P	ED	Pj	To
1.O contexto para Engenharia de Software 1. Visão geral de engenharia de software 2. Engenharia de sistemas baseados em computador 3. Software: engenharia ou arte? 4. Crise do software 5. Métodos de desenvolvimento 6. Ferramentas CASE, ambientes de desenvolvimento de software 7. Qualidade de software 8. Modelos de qualidade: CMM, ISO9000, SPICE 9. Código de Ética Profissional .1 10. Impactos Sociais	4h	0h	0h	0h	4h
2.Processos de desenvolvimento de software 1. O processo de desenvolvimento de software 2. Extração, análise e especificação de requisitos do software 3. O ciclo fundamental de desenvolvimento do software 4. Validação de requisitos, validação do produto final 5. Ciclo em cascata 6. Ciclo de prototipação 7. Ciclo espiral 8. Ciclo de Usabilidade 9. Desenvolvimento em fases: incremental e interativa .1 10. Avaliação comparativa dos ciclos propostos	6h	0h	0h	0h	6h
3.Análise e especificação de requisitos de software 1. Modelagem de sistemas com a visão de orientação por objetos 2. A linguagem UML e as visões que proporciona 3. Análise e especificação de requisitos de software 4. Modelagem de requisitos funcionais com Casos de Uso 5. Especificação de Casos de Uso: Diagramas e descrição textual 6. Especificação de requisitos não-funcionais 7. Validação de requisitos 8. O conceito de rastreabilidade de requisitos	8h	0h	0h	0h	8h
4.Modelagem estática de sistemas 1. Classes, objetos, classes abstratas 2. Modelagem de classes 3. Atributos, operações 4. Relações 5. Diagramas de classe 6. Herança e polimorfismo 7. Agregação e composição 8. Generalização 9. Classes persistentes, classes de negócio e classes de implementação .1 10. Rastreabilidade: relação das classes com os casos de uso .1 11. Implementação do diagrama de classes	12h	0h	0h	0h	12h

A autenticidade deste documento pode ser conferida no site <https://siadoc.ufv.br/validar-documento> com o código: 2SFM.JK2D.5NS9

5. Modelagem dinâmica de sistemas 1. Colaborações 2. Diagramas de sequência, colaboração, atividades e de estado 3. Técnicas de modelagem e especificação 4. Complementação do diagrama de classes com novas operações, atributos e classes 5. Relação entre o modelo dinâmico e o modelo estático do sistema 6. Relação com os casos de uso e requisitos funcionais 7. Rastreabilidade	12h	0h	0h	0h	12h
6. Modelagem física 1. Especificação de componentes: diagrama de componentes 2. Implantação física: diagrama de distribuição 3. Extensões para sistemas de tempo real	4h	0h	0h	0h	4h
7. Aspectos de implementação 1. Rastreabilidade 2. Padrões de projeto 3. Padrões de implementação 4. Padrões de Arquitetura 5. Mapeamentos de estruturas UML em linguagem de programação 6. Implementação de persistência 7. Mapeamento em banco de dados e bancos de objetos 8. Uso de componentes 9. Especificação de componentes .1 10. Construção de sistemas usando componentes .1 11. Ferramentas CASE	8h	0h	0h	0h	8h
8. Introdução à gerência de projetos de software 1. Gerência de projetos 2. Cronogramas e estimativas de tempo e custo 3. Organização e gerência de equipes 4. Métodos de estimativas de custo: COCOMO e Análise por Ponto-Função	6h	0h	0h	0h	6h
Total	60h	0h	0h	0h	60h

(T)Teórica; (P)Prática; (ED)Estudo Dirigido; (Pj)Projeto; Total(To)

Planejamento pedagógico	
Carga horária	Itens
Teórica	Apresentação de conteúdo oral e escrito com o apoio de equipamento (projutor, quadro-digital, TV, outros); Apresentação de conteúdo oral e escrito em quadro convencional; Debate mediado pelo professor; e Seminários
Prática	Não definidos
Estudo Dirigido	Não definidos
Projeto	Não definidos
Recursos auxiliares	Não definidos

INF 323 - Engenharia de Software II

Bibliografias básicas

Descrição	Exemplares
WAZLAWICK, R. S. Análise e design orientados a objetos para sistemas de informação, 3. ed. São Paulo, SP: Campus-Elsevier, 2014, 488p.	0
SOMMERVILLE, I. Engenharia de Software. 10. ed. Pearson Education, 2019. 768p.	0
BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro, RJ: Campus-Elsevier, 2007, 392p.	4

Bibliografias complementares

Descrição	Exemplares
BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML - GUIA DO USUÁRIO. Rio de Janeiro, RJ: Elsevier, 2005, 552p.	0
ERIKSSON, H.E.; PENKER, M. UML Toolkit. New York, NY, John Wiley & Sons, 1998.	0
FOWLER, M.; SCOTT, K. UML essencial: Um breve guia para a linguagem-padrão de modelagem de objetos, 3. ed. Porto Alegre, RS, Bookman, 2005.	6
GAMMA, E. e outros. Padrões de Projeto. Porto Alegre, RS, Bookman, 2000.	3
LARMAN, C. Utilizando UML e padrões. Porto Alegre, RS, Bookman, 2000.	1
PAULA-FILHO, W. P. Engenharia de Software: fundamentos, métodos e padrões. 3 ed. Rio de Janeiro: LTC, 2010. 1248 p. ISBN 9788521616504 (broch.).	5
PFLEEGER, S.L. Engenharia de Software: Teoria e prática. 2. ed. Pearson Education, 2004, 560p.	9
PRESSMAN, R. S. Engenharia de software: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, 2016. 968p. (broch.).	0
ROSENBERG, D.; SCOTT, K. Use-case driven object modeling with UML: a practical approach. Reading, Massachussets, Addison-Wesley, 1999	0
AMBLER, S. W. The Object Primer: Agile model-driven development with UML 2.0, Third Edition. Cambridge University Press, 2004.	0
AMBLER, S. W. Modelagem Ágil: práticas eficazes para programação extrema e o processo unificado. Porto Alegre, RS: Ed. Bookman, 2004.	4
VAZQUEZ, C. E. et al. Análise de Pontos de Função - Medição, Estimativas e Gerenciamento de Projetos de Software. 13. ed. Érica, 2013.	0