

Programa Analítico de Disciplina

INF 323 - Engenharia de Software II

Departamento de Informática - Centro de Ciências Exatas e Tecnológicas

Catálogo: 2020

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.

Ementa

O contexto para Engenharia de Software. Processos de desenvolvimento de software. Análise e especificação de requisitos de software. Desenvolvimento ágil de software. Modelagem dinâmica e modelagem física de sistemas. Modelagem estática de sistemas. Tópicos Avançados.

Pré e co-requisitos

INF 221

Oferecimentos obrigatórios

Não definidos

Oferecimentos optativos

| Curso | Grupo de optativas |
|-----------------------|--------------------|
| Ciência da Computação | Geral |

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 9.Código de Ética Profissional 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 10.Avaliação comparativa dos ciclos propostos 11.Processo Unificado 12.Modelos de Maturidade de processo 13.Gerência de Riscos | 18h | 0h | 0h | 0h | 18h |
| 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 9.Planejamento de ciclos iterativos 10.Cronograma de desenvolvimento | 6h | 0h | 0h | 0h | 6h |
| 4.Desenvolvimento ágil de software 1.Manifesto ágil para desenvolvimento de software 2.Metodologias ágeis 3.Customização de processos ágeis 4.Análise comparativa entre metodologias ágeis e disciplinadas 5.Combinação de metodologias ágeis com modelos de qualidade 6.Ferramentas de apoio ao processo de desenvolvimento ágil | 12h | 0h | 0h | 0h | 12h |

A autenticidade deste documento pode ser conferida no site <https://siadoc.ufv.br/validar-documento> com o código: VF6J.U6H5.FRTB

| | | | | | |
|---|------------|-----------|-----------|-----------|------------|
| 7. Estórias de usuário | | | | | |
| 5. Modelagem dinâmica e modelagem física de sistemas 1. Expansão de casos de uso 2. Operações e consultas de sistema 3. Diagramas de sequência, comunicação, atividades e de estado 4. Técnicas de modelagem e especificação 5. Complementação do diagrama de classes com novas operações, atributos e classes 6. Relação entre o modelo dinâmico e o modelo estático do sistema 7. Relação com os casos de uso, requisitos funcionais e rastreabilidade 8. Especificação de componentes: diagrama de componentes 9. Implantação física: diagrama de distribuição | 8h | 0h | 0h | 0h | 8h |
| 6. 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 10. Classes modais 11. Rastreabilidade: relação das classes com os casos de uso | 8h | 0h | 0h | 0h | 8h |
| 7. Tópicos avançados 1. Gerência de projetos: modelo PMBOK 2. Contratos de operação e consulta de sistema | 4h | 0h | 0h | 0h | 4h |
| 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 (projektor, quadro-digital, TV, outros); Apresentação de conteúdo oral e escrito em quadro convencional; Debate mediado pelo professor; e Seminários |
| Prática | <i>Não definidos</i> |
| Estudo Dirigido | <i>Não definidos</i> |
| Projeto | <i>Não definidos</i> |
| Recursos auxiliares | <i>Não definidos</i> |

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 |
| 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 |
| 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, Addisson-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 |
| KRUCHTEN, P. Introdução ao RUP: Rational Unified Process. Rio de Janeiro: Editora Ciência Moderna, 2003, 272p. | 0 |
| BROD, C. SCRUM – Guia prático para projetos ágeis. 2. Ed. Novatec, 2015, 200p. | 0 |
| SBROCCO, J. H. T. C.; MACEDO, P. C. Metodologias ágeis: Engenharia de software sob medida. Novatec, 2012, 256p. | 0 |