

MO620 – Engenharia de Software II
MC976 – Tópicos em Engenharia de Software II

Horário: terças e quintas 10:00 – 12:00 --- 2S 2022

Professora: Cecília Mary Fischer Rubira

cmrubira@ic.unicamp.br IC-Unicamp

Versão 3 16-9-2022

Disciplina Presencial: ensino presencial, vídeo-aulas disponibilizadas na plataforma, entrega de exercícios para avaliação e seminário.

Ferramentas utilizadas: Plataforma Moodle, Google Meet para encontros online, Diário de Bordo organizado por semanas.

Ementa:

- Conceitos da análise OO e projeto OO com foco na engenharia de modelagem de sistemas de software.
- Modelagem estática: construção de modelos de classes.
- Modelagem dinâmica: construção de modelos dinâmicos: modelos de sequência, de colaboração e de atividades.
- Conceitos Básicos: identificar objetos e classificá-los em classes, especificar atributos e operações das interfaces públicas (TAD e encapsulamento), herança e polimorfismo, e tratar relacionamentos de generalização/especialização, agregação e associação entre as classes.
- Conceitos Avançados: herança múltipla, interfaces, pacotes, metaclasses, delegação.

- Ênfase em Padrões de projeto: model-view-controller (MVC), composite, observer, singleton, abstract factory, etc
- Conceitos básicos de arquitetura de software, definição de componentes, conectores e configurações arquiteturais.
- Projeto arquitetural e padrões arquiteturais (layer (camadas), centrado em dados, máquina virtual, etc)
- Visão geral de Atributos de qualidade (segurança, disponibilidade, etc.) associados ao modelo arquitetural.
- Métodos de desenvolvimento de software centrados na arquitetura.
- Tolerância a Falhas e Tratamento de Exceções.
- Reutilização de Software, Desenvolvimento baseado em componentes e Linhas de Produto de Software.

Objetivo:

Ao final da disciplina o aluno deve ser capaz de criar modelos usando os conceitos do modelo de objetos e deve ser capaz de, partindo da especificação de um problema, criar modelos (estáticos e dinâmicos) que representem uma solução para o problema. Também o aluno deve ser capaz de definir um modelo inicial da arquitetura de software do sistema alvo, de acordo com os requisitos de qualidade priorizados durante o projeto arquitetural.

Critério de Avaliação:

As atividades desenvolvidas serão :

- (1) Exercícios e
- (2) 1 Seminário de pesquisa ao final do curso (obrigatório).

M = soma das notas das atividades / #total das atividades

Entre 8,5 – 10,0 → conceito A

Entre 7,0 – 8,4 → conceito B

Entre 5,0 – 6,9 → conceito C

Menor que 5,0 → conceito D (reprovado)

Graduação:

M >= 5,0 aprovado

Cronograma Detalhado: cronograma semanal detalhado no Moodle.

Bibliografia:

I. Sommerville. Software Engineering, Addison-Wesley, 10th edition, 2015.

G. Booch, J. Rumbaugh and I. Jacobson. The Unified Modeling Language User Guide, second edition, Addison-Wesley, 2005.

E. Gamma et al., Design Patterns: Elements of reusable OO Software, Addison-Wesley, 1995.

F. Buschmann et al., A System of Patterns: Pattern-Oriented Software Architecture, Wiley, 1996.

L. Bass, P. Clements & R. Kazman, Software Architecture in Practice, Second Edition, Addison-Wesley, 2003, SEI Series in Software Engineering

C. Szyperski, Component software: beyond OO programming, Addison-Wesley, 1998.

M. Shaw & D. Garlan. Software Architecture: Perspectives on an emerging discipline, Prentice Hall, 1996

F. Bachman et al., The architecture-based design method, technical report CMU/SEI-2000-TR-001, Software Engineering Institute, Carnegie-Mellon University, January 2000.

P. Krutchen. The 4+1 view model of software architecture. IEEE Software, pages 42–50, November 1995.

M. Barbacci et al., Steps in architecture tradeoff analysis method: quality attribute models and analysis. technical report CMU/SEI-97-TR-029, Software Engineering Institute, Carnegie-Mellon University, May 1998.