

## Plano de Desenvolvimento da Disciplina

### Ementa

- Paradigmas de Engenharia de Software.
- Modelos de Processos de Desenvolvimento de Software.
- Levantamento de requisitos.
- Técnicas e ferramentas de especificação de requisitos.
- Métodos de análise e projeto de sistemas de software.
- Implementação de sistemas de software.

### Programa

1. Fundamentos
  - Visão geral da área de engenharia de software;
  - Conceitos de produto (sistemas de software) e processos de desenvolvimento de software;
  - Modelos de processos de desenvolvimento de software.
2. Extração de requisitos de sistemas de software
  - Conceitos;
  - Técnicas para extração de requisitos;
3. Análise de sistemas de software
  - Análise Orientada a Objetos.
  - Modelos de casos de uso, modelos de classes, modelos comportamentais (modelos de comunicação e de seqüência) em UML.
4. Projeto de sistemas de software
  - Aspectos Fundamentais: abstração, refinamento, modularidade, componentização.
  - Modelos de projeto.
  - Projeto Arquitetural e Arquitetura de software;
  - Padrões arquiteturais e padrões de projeto;
5. Projeto detalhado de sistemas de software
  - Qualidade do projeto: coesão, acoplamento e outros;

## **Critério de Avaliação**

- 1 prova escrita individual (prova vale 5.0 pontos)
- 3 atividades práticas em equipe (ativ1 e ativ2 valem 1.0 ponto cada, e ativ3 vale 2.0 pontos)
- participação em sala e entrega de listas de exercícios (participa vale 1.0 ponto)

$M = \text{prova} + \text{ativ1} + \text{ativ2} + \text{ativ3} + \text{participa}$

Data da prova: 21 de novembro de 2016, segunda.

- se  $M \geq 5,0$  o aluno está aprovado;
- caso contrário, o aluno está de exame.

## **Exame**

Alunos com pelo menos 75% de frequência poderão realizar o exame.

Média do Exame =  $(M + \text{Nota da Prova de Exame})/2$

Data do Exame: 21 de dezembro de 2016, quarta.

## **Horário de Atendimento com Professor**

A professora vai tirar dúvidas ao final da aula e o aluno pode enviar uma mensagem para o professor (cmrubira@ic.unicamp.br) solicitando o agendamento de atendimento.

## **Datas Importantes**

- Data da Prova: 21 de novembro de 2016, segunda.
- Data do Exame: 21 de dezembro de 2016, quarta.

## **Temas Sugeridos para o Projeto**

Um sistema de gerenciamento de informações para uma empresa desenvolvimento de software (CONPEC).

## **Páginas importantes**

Ensino Aberto <https://www.ggte.unicamp.br/ea/>

Links dos Livros do Ian Sommerville:

10th ed: <http://iansommerville.com/software-engineering-book/>

9th ed: <http://www.softwareengineering-9.com/>

8th ed: <http://ifs.host.cs.st-andrews.ac.uk/Books/SE8/>

# Bibliografia Recomendada

## Engenharia de Software

- I. Sommerville, *Software Engineering*, Addison-Wesley, 10th edition, 2015.
- I. Sommerville, *Software Engineering*, Addison-Wesley, 9th edition, 2011.
- I. Sommerville, *Software Engineering: (Update) (8th edition)*, Addison Wesley, 8th edition, 2007.
- Roger S. Pressman, *Software Engineering A Practitioner's Approach* 7th Edition, Roger S Pressman, R. S. Pressman & Associates, Inc., ISBN: 0073375977, Copyright year: 2010
- Roger S. Pressman, *Engenharia de Software*, 6a. edição, 2006, McGraw-Hill, São Paulo.
- Jalote, Pankaj. *A Concise Introduction to Software Engineering*. London: Springer-Verlag London, 2008. v.: digital. (Undergraduate topics in computer science). ISBN 9781848003026. E-BOOK disponível em: <http://dx.doi.org/10.1007/978-1-84800-302-6> (Acessado em: 12/8/2013)
- Ghezzi, C., Jazayeri, M. & Mandrioli, D., *Fundamentals of Software Engineering*, Prentice Hall, 1991.
- E.J. Braude, *Software Engineering: An OO Perspective*, John Wiley & Sons, INC., 2001.
- W.P. Paula Filho, *Engenharia de Software: Fundamentos, Métodos e Padrões*, Segunda Edição, Editora LTC, 2001.

## Modelagem e Projeto Orientados a Objetos

- Rumbaugh, J. et al., *Object-Oriented Modeling and Design*, Prentice Hall, 1991.
- Booch, G., *Object-Oriented Design with Applications*, Benjamin-Cummings, 1991.
- G. Booch, J. Rumbaugh & I. Jacobson, *The Unified Modeling Language User Guide*, Addison Wesley, 1999.
- H. Eriksson & M. Penker, *UML Toolkit*, Wiley, 1998.
- P. Muller, *Instant UML*, Wrox, 1997.
- C. Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*, Second Edition, Prentice-Hall, 2002.
- P. Harmon & M. Watson, *Understanding UML*, Morgan Kaufmann, 1998.
- C. Richter, *Designing Flexible OO Systems with UML*, MTP, 2001.
- M. Page-Jones, *Fundamentals of OO Design in UML*, Addison-Wesley, 2000.
- P. Stevens, *Using UML: software engineering with objects and components*, Addison-Wesley, 1999.
- Y. Lau, *The Art of Objects: OO design and architecture*, Addison-Wesley, 2001.
- A. Carvalho & T. Chiossi, *Introdução à Engenharia de Software*, Editora da UNICAMP, 2001.
- C.M.F. Rubira, Apostila *Introdução à Análise Orientada a Objetos e ao Projeto Arquitetural*, IC-UNICAMP, 2014.

## Metodologias OO

- I. Jacobson, G. Booch & J. Rumbaugh, *The Unified Software Development Process*, Addison Wesley, 1999.
- Krütchen, P., *The Rational Unified Process: An Introduction*, Second Edition, Addison-Wesley, 2000.
- J. Cheesman & J. Daniels, *UML Components: A Simple Process for Specifying Component-Based Software*, Addison-Wesley, 2001.
- D. D'Souza & A. Wills, *Objects, Components and frameworks with UML: The Catalysis Approach*, Addison-Wesley, 1999.
- S.R. Palmer & J.M. Felsing, *A Practical Guide to Feature-driven development*, The Coad Series.

## Linguagens de Programação OO

- K. Arnold & J. Gosling, *The Java Programming Language*, second edition, Addison-Wesley, 1997.
- Flanagan, D. *Java in a Nutshell*, O'Reilly & Associates, 1996.
- K. Beck, *Extreme Programming: embrace change*, Addison-Wesley, 2000.

## Casos de Uso

- G. Schneider & J.P. Winters *Applying Use Cases: A Practical Guide*, Second Edition, Addison-Wesley, 2001.
- A. Cockburn *Writing Effective Use Cases*, Addison-Wesley, 2001.
- D. Kulak & E. Guiney *Use Cases: Requirements in Context*, Addison-Wesley, 2000.
- I. Jacobson, *OO Software Engineering: A Use Case Driven Approach*, Addison-Wesley, 1992.

## Design Patterns

- M. Fowler, *Analysis Patterns: Reusable Object Models*, Addison-Wesley, 1997.
- M. Grand, *Patterns in Java*, Wiley, 1998.
- E. Gamma et al., *Design Patterns: Elements of reusable OO Software*, Addison-Wesley, 1995.
- Pattern Languages of Program Design, 1, 2, 3 e 4, Software Patterns Series.

## Frameworks Orientados a Objetos

- W. Pree, *Design Patterns for OO Software Development*, Addison-Wesley, 1995.
- M. Fayad & R. E. Johnson, *Domain-Specific Application Frameworks*, Wiley, 2000.
- M. Fayad, D. C. Schmidt & R.E. Johnson, *Implementing Application Frameworks*, Wiley, 1999.
- M. Fayad, D. C. Schmidt & R.E. Johnson, *Building Application Frameworks*, Wiley, 1999.

## Arquitetura de Software

- 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. Hofmeister et al., *Applied Software Architecture*, Addison-Wesley, 2000.
- M. Shaw & D. Garlan. *Software Architecture: Perspectives on an emerging discipline*, Prentice Hall, 1996.
- J. Bosch, *Design and use of software architecture*, Addison-Wesley, 2000.
- L. Barroca et al. *Software Architectures: advances and applications*, Springer, 2000.
- L. Hohmann, *Beyond Software Architecture: creating and sustaining winning solutions*, Addison-Wesley, 2003.
- P.C.Clements & L.Northrop. *Software Architecture: an executive overview*. technical report CMU/SEI-96-TR-003, Software Engineering Institute, Carnegie-Mellon University, February 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.

## Desenvolvimento Baseado em Componentes

- C. Szyperski, *Component software: beyond OO programming*, Addison-Wesley, 1998.
- G.T.Leavens & M.Sitaraman, *Foundations of Component-based Systems*, Cambridge University press, Cambridge, UK, 2000.
- F. Bachman et al., *Volume II: Technical concepts of component-based software engineering*, technical report CMU/SEI-2000-TR-008, Software Engineering Institute, Carnegie-Mellon University, April 2000.
- J. Sametinger, *Software Engineering with reusable components*, Springer, 1997.