MC426 – Engenharia de Software
MC436 – Introdução à Engenharia de Software
Instituto de Computação — UNICAMP
2o. Semestre de 2013
Profa. Cecília M. F. Rubira
sala 13, cmrubira@ic.unicamp.br
PED: Gustavo Waku
sala 91 (LSD), gustavo.waku@gmail.com

Plano de Desenvolvimento da Disciplina

atualizado em 13/ago/2013

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.Introdução
- 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;
- Projeto detalhado de sistemas;
- Qualidade do projeto: coesão, acoplamento e outros;

```
5. Gerenciamento do desenvolvimento de projetos de software:
      - Definição de metas e objetivos;
      - Estimativas: métricas e modelos de custo;
      - Estudo de viabilidade;
      - Cronogramas.
```

Critério de Avaliação

- 2 provas escritas individuais (cada prova valendo 3.5 pontos)
- 2 atividades práticas em equipe (cada atividade valendo 1.5 pontos)

  M = prova1 + prova2 + ativ1 + ativ2

  Data da 1a. prova: 25 de setembro de 2013, quarta.

  Data da 2a. prova: 13 de novembro de 2013, quarta.

- se M >= 5,0 o aluno está aprovado;
- caso contrário, o aluno está de exame.

Exame

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

Média do Exame = (M + Nota da Prova de Exame)/2

Data do Exame: 9 de dezembro de 2013, segunda.

Horário de Atendimento

Enviar uma mensagem para o professor solicitando o agendamento de atendimento, assim como para o PED. Combinar um horário fixo adequado para os estudantes.

Datas Importantes

- Data da 1a. prova: 25 de setembro de 2013, quarta.
- Data da 2a. prova: 13 de novembro de 2013, quarta.
- Data do Exame: 9 de dezembro de 2013, segunda.

Bibliografia Recomendada

Engenharia de Software

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*, IC-UNICAMP, 2003.

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-Wesly, 1995.

M. Fayad & R. E. Johnson, *Domain-Specific Application Framewoks*, Wiley, 2000.

M. Fayad, D. C. Schmidt & R.E. Johnson, *Implementing Applcation Frameworks*, Wiley, 1999.

M. Fayad, D. C. Schmidt & R.E. Johnson, *Building Applcation 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, Carnigie-Mellon University, February 1996.

F. Bachman et al., *The architecture-based design method*, technical report CMU/SEI-2000-TR-001, Software Engineering Institute, Carnigie-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, Carnigie-Mellon University, May 1998.

Desenvolvimento Baseado em Componentes

C. Szyperski, *Component software: beyond OO programming*, Addison-Wesly, 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, Carnigie-Mellon University, April 2000.

J. Sametinger, *Software Engineering with reusable components*, Springer, 1997.

Páginas importantes

- Cecília M.F. Rubira [http://www.ic.unicamp.br/∼cmrubira]
- Ensino Aberto [http://www.unicamp.br/ea/]
- Páginas dos Livros do Sommerville 8th edition: http://ifs.host.cs.st-andrews.ac.uk/Books/SE8/ (acessado em 12/8/2013)

  9th edition: http://www.softwareengineering-9.com/ (acessado em 12/8/2013)