



Universidade Federal de Santa Catarina
Centro Tecnológico
Departamento de Informática e Estatística
Programa de Pós-Graduação em Ciência da Computação



Plano de Ensino

1) Identificação

Disciplina: INE410117 - Modelagem de Sistemas Orientados a Objetos

Carga horária: 45 horas/aula – 3 créditos

Professor: Raul Sidnei Wazlawick

2) Requisitos: não há.

3) Ementa:

Modelagem de negócio. Modelagem de requisitos de sistema. Modelagem conceitual. Modelagem funcional. Modelagem dinâmica. Modelagem de interface. Modelagem de persistência.

4) Objetivos:

- **Geral:** Capacitar o aluno a entender e praticar análise de design orientados a objetos com técnicas avançadas de modelagem.

- **Específicos:**

- Capacitar em modelagem de negócio.
- Capacitar em casos de uso e requisitos.
- Capacitar em modelagem conceitual.
- Capacitar em planejamento de projeto.
- Capacitar em modelagem funcional e dinâmica.
- Capacitar em modelagem de interface e persistência

5) Conteúdo Programático:

- Introdução [4 horas-aula]
 - Desenvolvimento OO
 - UML
 - UP
- Modelagem de negócio [4 horas-aula]
 - Casos de uso de negócio
 - Atividades de negócio
 - Aspectos de negócio dependentes de estado
- Requisitos em alto nível [4 horas-aula]
 - Atores de sistema
 - Casos de uso de sistema
 - Requisitos
 - Modelo conceitual preliminar
- Planejamento baseado em casos de uso [4 horas aula]
 - Análise de pontos de caso de uso.

- ii. Planejamento de projeto com ciclos iterativos.
- e. Casos de uso detalhados [4 horas-aula]
 - i. Fluxo principal
 - ii. Fluxos alternativos
 - iii. Casos de uso incluídos
 - iv. Diagramas de sequencia de sistema
- f. Modelagem conceitual [8 horas-aula]
 - i. Fundamentos
 - ii. Padrões de análise
- g. Modelagem funcional com contratos OCL [4 horas-aula]
 - i. Precondições
 - ii. Pos condições
 - iii. Exceções
 - iv. Retornos
- h. Modelagem dinâmica com diagramas de comunicação [4 horas-aula]
 - i. Responsabilidades e operações básicas
 - ii. Visibilidade
 - iii. Realização dinâmica das pós condições
 - iv. Consultas de sistema
 - v. Diagrama de classes de design
- i. Geração de código [2 horas-aula]
 - i. Classes e atributos
 - ii. Associações
 - iii. Métodos delegados e operações de sistema
- j. Teste [2 horas-aula]
 - i. Teste funcional de unidade
 - ii. Teste de operação de sistema
 - iii. Teste de caso de uso
- k. Modelagem de interface [3 horas aula]
 - i. WebML
 - ii. Prática com WebRatio
- l. Persistência [2 horas-aula]
 - i. Equivalência objeto-relacional
 - ii. Framework de persistência

6) Metodologia:

- Aulas expositivas
- Trabalhos práticos
- Apresentação e discussão de projetos

7) Avaliação:

- Realização de trabalho prático ao longo da disciplina.

8) Cronograma:

- Semana 1: Introdução
- Semana 2: Modelagem de negócio
- Semana 3: Requisitos em alto nível
- Semana 4: Planejamento baseado em casos de uso
- Semana 5: Casos de uso detalhados

- Semana 6: Modelagem conceitual
- Semana 7: Modelagem conceitual
- Semana 8: Modelagem funcional com contratos OCL
- Semana 9: Modelagem dinâmica com diagramas de comunicação
- Semana 10: Geração de código
- Semana 11: Teste
- Semana 12: Modelagem de interface
- Semana 13: Modelagem de interface
- Semana 14: Persistência
- Semana 15: Apresentação final de trabalhos

9) Bibliografia básica:

- Wazlawick, R. S. Análise e projeto de sistemas de informação orientados objetos, 2ª edição, Rio de Janeiro, Elsevier, 2011.
- WAZLAWICK, R. S. Engenharia de software: conceitos e práticas, Rio de Janeiro, Elsevier, 2013.

10) Bibliografia complementar:

- Ambler, S. Process patterns. Cambridge University Press, 1998.
- Ambler, S., Constantine, L., Smith, R. The Unified Process elaboration phase: best practices in implementing the UP. CMP Books, 2000.
- Ambler, S., Jeffries, R. Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process. John Willey and Sons, 2002.
- Anda, B. Estimating Software Development Effort Based on Use Cases – Experiences from Industry. 4th International Conference on the Unified Modelling Language. Toronto: Springer-Verlag, 2001. p. 487-502.
- Arlow, J., Neustadt, I. UML and the Unified Process: practical object-oriented analysis and design. Pearson Education, 2001.
- Beck, K. Test-Driven Development by Example. Addison Wesley, 2003.
- Beizer, B. Software Testing Techniques. 2. ed. Van Nostrand Reinhold, 1990.
- Bezerra, E. Princípios de análise e projeto de sistemas com UML. Campus–Elsevier, 2003.
- Boehm, B.W. (2000). Software Cost Estimation With COCOMO II. Prentice-Hall, 2000.
- Booch, G. Object-oriented analysis and design with applications. Addison-Wesley, 1993.
- Booch, G. Object solutions □ managing the object-oriented project. Addison-Wesley, 1996.
- Braz, M., Vergilio, S. Software Effort Estimation Based on Use Cases. Proceedings of the 30th Annual International Computer Software and Applications Conference, 2006. p. 221-28.
- Cabot, J. (2007). From Declarative to Imperative UML/OCL Operation Specifications. Lecture Notes in Computer Science, 198-213.
- Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S., Matera, M. Designing data-intensive Web applications. Morgan Kaufmann Publishers, 2003.
- Chimiak–Opoka, J., et al. (2011). OCL Tools Report based on the IDE4OCL Feature Model. Electronic Communications of the EASST, 44.
- Cox, B. Object-oriented programming: an evolutionary approach. Addison-Wesley, 1986.
- D’Souza, D.F., Wils A.C. Objects, components, and frameworks with UML. Addison-Wesley, 1999.
- Dustin, E. Automate Regression Tests When Feasible. In: E. Dustin. Effective Software Testing: 50 Specific Ways to Improve Your Testing. Addison-Wesley Professional, 2002.
- Emam, K., Drouin, J.N., Melo, W. Spice: the theory and practice of software process improvement and capability determination. IEEE Computer Society, 1998.

- Embley, D.W., Kurtz, B.D., Woodfield, S.N. Object-oriented systems analysis: a model-driven approach. Prentice-Hall, 1992.
- English, A. V. Business Modeling with UML: Understanding the similarities and differences between business use cases and system use cases. Disponível em: <http://www.ibm.com/developerworks/rational/library/apr07/english/>. Consultado em: 14 de novembro de 2012.
- Erickson, H.E., Penker, M. UML toolkit. John Wiley and Sons Inc., 1998.
- Fowler, M., Scott, K. UML distilled. Addison-Wesley, 1997.
- Fowler, M. Patterns of enterprise application architecture. Addison-Wesley, 2003.
- Gamma, E., Helm, R., Johnson, R., Vlissides, J. Design patterns: Elements of reusable object-oriented software. Addison-Wesley, 1995.
- Jacobson, I., Christerson, M., Jonsson, P., Övergaard, G. Object-oriented software engineering □- a use CASE driven approach. Addison-Wesley, 1992.
- Jacobson, I. The object advantage □- business process reengineering with object technology. Addison-Wesley, 1994.
- Jacobson, I., Booch, G., Rumbaugh, J. The unified software development process. Addison-Wesley, 1999.
- Jacobson, I., Spence, I., Bittner, K. Use Case 2.0 – The Guide to Succeeding with Use Cases. Ivan Jacobson International, 2011. Ebook disponível em: http://www.ivarjacobson.com/use_case2.0_ebook/
- Kamal, M. W. & Ahmed, M. A. A Proposed Framework for Use Case Based Effort Estimation using Fuzzy Logic: Building Upon the Outcomes of a Systematic Literature Review. International Journal on New Computer Architectures and Their Applications, 1(4), 2011, p. 976-99.
- Karner, G. Use CASE points □ resource estimation for objectory projects. Objective Systems, 1993.
- Kroll, P., MacIsaac, B. Agility and Discipline Made Easy: Practices from OpenUP and RUP. Addison-Wesley Professional, 2006.
- Kroll, P., Kruchten, P. The Rational Unified Process Made Easy: A Practitioner’s Guide to the RUP. Addison Wesley, 2003.
- Kruchten, P. The rational unified process: an introduction. Addison-Wesley, 2000.
- Kruchten, P. The rational unified process made easy: a practitioner’s guide to rational unified process. Addison-Wesley, 2003.
- Larman, C. Applying UML and patterns: an introduction to object-oriented analysis and design and the unified process. 3rd Edition. Prentice Hall, 2004.
- Lieberherr, Karl, Holland, I. Assuring good style for object-oriented programs. IEEE Software: 38–48, September 1989.
- Maldonado, J.C., Delamaro, M.E., Jino, M. Introdução ao teste de software. Campus-Elsevier, 2007.
- Meyer, B. Object-oriented software construction. Prentice Hall, 1988.
- Meyer, B. Eiffel: the language. Prentice-Hall, 1992.
- Miles, R., Hamilton, K. Learning UML 2.0. O’Reilly, 2006.
- Mitchel, R., McKim, J. Design by contract by example. Addison-Wesley, 2001.
- Mohagheghi, P., Anda, B., Conradi, R. Effort Estimation of Use Cases for Incremental Large Scale Software Development. International Conference on Software Engineering, 2005.
- Myers, G. J., Sandler, C., Badgett, T., Thomas, T. M. The Art of Software Testing. 2. ed. New Jersey: John Wiley & Sons, 2004.
- OMG - Object Management Group. OCL 2.3.1 Specification (2010).
- Page-Jones, M. Fundamentos do desenho orientado a objeto com UML. Makron Books, 2001. (Tradução de Fundamentals of object-oriented design in UML.)

- Paula Filho, W.P. Engenharia de software: fundamentos, métodos e padrões. LTC, 2003).
- Pereira e Silva, R. UML 2 □ Modelagem orientada a objetos. Visual Books, 2007.
- Pereira e Silva, R. Como modelar com UML 2. Visual Books, 2009.
- Pressman, R.S. Software engineering: a practitioner's approach. McGraw Hill, 2010.
- Ribu, K. Estimating Object-Oriented Software Projects with Use Cases. Master of Science Thesis, University of Oslo, Department of Informatics, 2001.
- Riel, A. J. Object-oriented design heuristics. Addison-Wesley, 1996.
- Rumbaugh, J., Blaha, M.R., Lorensen, W., Eddy, F., Premerlani, W. Object-oriented modeling and design. Prentice Hall, 1990.
- Rumbaugh, J., Jacobson, I., Booch, G. The Unified Modeling Language reference manual. Addison-Wesley, 1999.
- Santos, C.C. Geração automática de diagramas de comunicação a partir de contratos OCL. Dissertação de Mestrado, UFSC-PPGCC, 2007.
- Satzinger, J. W., Jackson, R. B., Burd, S. D. Systems Analysis and Design in a Changing World. Course Technology, 6th edition, 2011.
- Schneider, G., Winters, J. P. Applying Use Cases: A Practical Guide. Addison Wesley, 1998.
- Scott, K. The Unified Process explained. Addison-Wesley Pub Co., 2001 (O processo unificado explicado. Bookman, 2003.)
 - Shalloway, A., Trott, J.R. Explicando padrões de projeto: uma nova perspectiva em projeto orientado a objeto. Bookman. (Tradução de Design patterns explained: a new perspective on object-oriented design.)
 - Silva, A.A., Gomide, C.F., Petrillo, F. Metodologia e projeto de software orientados a objetos: modelando, projetando e desenvolvendo sistemas com UML e componentes distribuídos. Érica, 2003.
 - Sommerville, I. Software Engineering, 7th ed. Harlow, UK: Addison Wesley, 2006.
 - Warmer, J., Keppe, A. The Object Constraint Language: precise modeling with UML. Addison-Wesley Pub Co., 1998.
 - Wirfs-Brock, R., McKean, A. Object design: roles, responsibilities, and collaborations. Addison-Wesley, 2002.