

Engenharia de Software N

Profa. Karin Becker

Instituto de Informática - UFRGS

Objetivos

- Compreender a importância, **contexto** e **conceitos** da Engenharia de Software;
- Compreender as **fases**, **atividades** e **papéis** envolvidos no **ciclo de vida** de um software ;
- Aplicar técnicas de **modelagem** para as principais atividades do ciclo de vida de um software;
- **Dentro de uma metodologia de desenvolvimento** de software, **relacionar** as principais **técnicas de modelagem** às **atividades do ciclo de vida** de um software;
- Conhecer tendências em Engenharia de Software.

Dinâmica

- Aulas expositivas
- Leituras complementares
- Exercícios práticos, laboratórios
- Dinâmicas permitindo compreender alguns aspectos da Engenharia de Software e do processo de desenvolvimento de software
- Trabalho prático EM GRUPO
 - Aplicação do conteúdo
 - Vivência de algumas situações
 - Posicionamento crítico

Comunicação Extra-classe

- Moodle
 - Alunos devem se inscrever na disciplina
 - Moodle concentra todo o material da disciplina (transparências, exercícios, atividades extra-classe, etc)
 - Todos avisos extra-classe são feitos via moodle
 - Procure comunicar-se via o moodle com o professor
 - Moodle registra toda comunicação
 - Os fóruns de dúvidas permitem que os alunos troquem idéias, compartilhem dúvidas comuns, etc
- Gabinete de trabalho
 - Prédio 73, sala 242
 - Política “portas-abertas”
 - se quiser ter atendimento garantido, procure combinar um horário com antecipação

Bibliografia

- **BIBLIOGRAFIA BÁSICA**

- Sommerville, I. Engenharia de software, Pearson. (atualmente na 9 edição – biblioteca possui até a 8 edição)
- Larman, C. Utilizando UML e Padrões - Uma Introdução à Análise e ao Projeto Orientados a Objetos, Bookman. (atualmente na 3 edição – as edições anteriores são mais enxutas)
- Ambler, S., Modelagem Ágil, Bookman.

- **Bibliografia Complementar**

- Ambler, S., The elements of UML 2.0 Style. Cambridge University Press, 2005.
- Beck, K. Programação eXtrema (XP) Explicada, Bookman, 2004.
- Cockburn, A. Escrevendo Casos de Uso Eficazes, Bookman, 2004.
- Pressman, Roger. Engenharia de Software. Ed. Makron Books, 2006.
- Humphrey, Watts S. A Discipline for Software Engineering. Ed. Addison Wesley, 1995.
- Braude, E. Software Engineering – An Object-Oriented Perspective, John Wiley & Sons, 2001.
- Artigos selecionados

Avaliação

- 2 provas com conteúdo cumulativo
 - “cola autorizada”
- 1 trabalho prático dividido em etapas, e desenvolvido **em grupo**
 - Prática da teoria vista em aula no desenvolvimento de um sistema, com ênfase em modelagem, mas envolvendo também programação
 - Dividido em etapas
 - Simulação de situações possíveis em um ambiente de desenvolvimento de software, e vivência das suas consequências
 - Análise crítica
- Exercícios práticos, laboratórios, dinâmicas, participação em aula, maturidade e posicionamento crítico, presença (não limitada à física)

Avaliação

- $\text{notaFinal} = \text{Prova1} * 0,2 + \text{Prova2} * 0,20 + \text{Trabalho} * 0,5 + \text{Participação} * 0,1$
- O referencial usado para traduzir a notaFinal em conceito é
 - $A \cong [9 \text{ .. } 10]$
 - $B \cong [7,5 \text{ .. } 9)$
 - $C \cong [6 \text{ .. } 7,5)$
 - $D \cong [0 \text{ .. } 6)$
 - FF – frequência inferior a 75%

Avaliação - Recuperação

- Público alvo: alunos com conceito D
 - Ter realizado pelo menos uma prova
 - Ter entregue o trabalho prático
- Como: prova de recuperação versando sobre todo o conteúdo da disciplina.
- Resultado: Se a nota obtida na prova de recuperação for igual ou superior a 6,0, o conceito mudará para C.