

PLANO DE ENSINO

INF01022 – LABORATÓRIO DE SISTEMAS DE SOFTWARE

Semestre: 2009/2

Carga horária: 04 h

Créditos: 04

Professor(es): Marcelo Pimenta (mpimenta@inf.ufrgs.br) – Turma U

Súmula:

O conteúdo a ser abordado na disciplina é a consolidação, através de um projeto real de uma dada organização, dos conceitos abordados na parte básica dos curso, via desenvolvimento em grupo, sob a orientação do professor.

Conteúdo Programático:

O objetivo é capacitar o aluno a conceber, projetar, desenvolver e gerenciar o desenvolvimento de sistemas de software de complexidade mediana. Para alcançar este objetivo, a disciplina está montada sobre uma pequena revisão dos conceitos relacionados a desenvolvimento de sistemas e uma grande parte dedicada a estudos de caso, que serão feitos pelos alunos em grupos de trabalho. As atividades relacionadas a estudos de caso incluem: Acompanhamento do andamento dos trabalhos; Apresentação e discussão dos modelos já elaborados; Discussão e apresentação de problemas de arquiteturas de sistemas; Discussão e apresentação de alternativas de soluções aos problemas encontrados; Discussão dos problemas de padrões de software e sua aplicação a arquitetura do sistema em desenvolvimento; Discussão de problemas de gerenciamento de desenvolvimento e aplicação as atividades de cada grupo e cada aluno; Apresentação das soluções intermediárias e dos resultados finais obtidos.

Procedimento Didático:

As aulas serão de cunho teórico-expositivo intercaladas com discussões sobre os exercícios e os trabalhos extra-classe, além de aulas em laboratório, quando forem julgadas necessárias.

Método de Avaliação:

A avaliação será baseada principalmente nos trabalhos individuais e/ou em grupo realizados no decorrer da disciplina. Serão realizados no mínimo 2 (dois) trabalhos com data de entrega a ser fixada pelo professor. A cada trabalho será atribuído um conceito de A (melhor) a D (pior).

O conceito final do aluno será atribuído levando-se em consideração a participação do aluno nas atividades em classe e extra-classe (15%) e a média dos trabalhos realizados pelo aluno (85%). Em particular o conceito dos trabalhos não poderá ser menor que C para a obtenção de um conceito de aprovação.

RECUPERAÇÃO

Os alunos com conceito final insuficiente (menor que C) podem realizar uma Prova de Recuperação, com data e conteúdo a serem fixados pelo professor.

Bibliografia Básica:

1. COCKBURN, A. Escrevendo Casos de Uso Eficazes, Bookman, 2004.
2. MEYER, B. Object-Oriented Software Construction, 2nd Edition, 1997.
3. MEYER, B. Touch of Class - Learning to Program Well with Objects and Contracts, Springer-Verlag, 2009.
4. McCONNELL, Steve. Code Complete: A Practical Handbook of Software Construction. Microsoft Press, 2a. edição, Julho, 2004.
5. BRAUDE, Eric. Projeto de Software, Bookman, 2005.

PLANO DE ENSINO

6. Gamma, Erich et all: *Padrões de Projeto*, Porto Alegre: Bookman, 2000.
7. Larman, C. *Utilizando UML e Padrões - Uma Introdução à Análise e ao Projeto Orientados a Objetos*, Bookman, 2008.
8. Fowler, M. *Refatoração – Aperfeiçoando o Projeto de Código Existente*, Bookman, 2004.
9. Ambler, S. , *Modelagem Ágil*, Bookman, 2004.
10. Beck, K. *Programação eXtrema (XP) Explicada*, Bookman, 2004.
11. Braude, E. *Software Engineering – An Object-Oriented Perspective*, John Wiley & Sons, 2001.

Bibliografia Complementar:

12. HORSTMANN, C. *Object-Oriented Design & Patterns*, Joyhn Wiley & Sons, 2004.
13. BECK, K. *Implementation Patterns*, Pearson Education, Addison-Wesley, 2008.
14. Artigos de publicações como ACM Computing Surveys, Communications of ACM, IEEE Transactions on Software Engineering, IEEE Software, Revista Brasileira de Computação, entre outras.
15. Artigos recentes publicados em Conferências como International Conference on Software Engineering (ICSE), Simpósio Brasileiro de Engenharia de Software (SBES), entre outros.

Cronograma por aula: (4 horas-aula na 6a f, 8:30-12:10)

aula	Dia	Conteúdo
1	21/08	Apresentação da disciplina. Apresentação e discussão dos temas e formato dos estudos de caso. Formação de grupos. Revisão de Processos de Software, Sistemáticos (UP) e Métodos Ágeis.
2	28/08	Revisão de Planejamento, Análise e Engenharia de Requisitos, Exercícios e Leituras. Definição dos Trabalhos Práticos.
3	04/09	Revisão de Projeto e Implementação, Exercícios e Leituras. TODOS: Entrega de Lista de Requisitos e versão preliminar do Glossário;
4	11/09	Revisão de Patterns e Arquitetura de Software, Gerência de Versões e Configurações, Exercícios e Leituras. TODOS: Planejamento do processo, atividades e resultados
5	18/09	Aula de acompanhamento de trabalho; UP: Entrega dos Casos de Uso Preliminares e Modelo Conceitual Preliminar; Ágeis: planejamento de iterações e releases
6	25/09	Aula de acompanhamento de trabalho; Ágeis: 1o release – fim da primeira iteração
7	02/10	Aula de acompanhamento de trabalho Ágeis: 2o release – fim da segunda iteração
8	09/10	Aula de acompanhamento de trabalho; UP: Entrega dos Principais Casos de Uso Essenciais Ágeis: 3o release – fim da terceira iteração
9	16/10	Aula de acompanhamento de trabalho; UP: Entrega dos principais Casos de uso Reais ; Entrega do Design das Interfaces (croquis ou protótipos) Ágeis: 4o release – fim da quarta iteração
10	23/10	Aula de acompanhamento de trabalho Ágeis: 5o release – fim da quinta iteração
11	30/10	Aula de acompanhamento de trabalho;

PLANO DE ENSINO

UP: Entrega do Diagrama de Classes e diagramas de Projeto dos principais comportamentos;

Ágeis: 6o release – fim da sexta iteração

12 06/11 Aula de acompanhamento de trabalho

Ágeis: 7o release – fim da sétima iteração

13 13/11 Aula de apresentação/demonstração dos resultados dos trabalhos

14 20/11 Aula de apresentação/demonstração dos resultados dos trabalhos

15 27/11 **Recuperação**

- 04/12 Aula de reserva

- 11/12 Aula de reserva