

UNIVERSIDADE FEDERAL DE MINAS GERAIS / INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
Planejamento da disciplina: Engenharia de Software
Professor: Clarindo Isaías Pereira da Silva Pádua
Carga Horária: 60 horas, 4 créditos
Horário: 13:00 – 14:40 horas, sala 2015

1 Objetivos da Disciplina

Essa disciplina tem por objetivo apresentar uma visão geral de processos e técnicas de desenvolvimento de software utilizados para assegurar qualidade em produtos de software. A disciplina deverá tornar o aluno apto a:

- conhecer um processo típico de produção de software;
- conhecer padrões e técnicas no estado da arte para:
 - engenharia de requisitos;
 - análise orientada a objetos;
 - desenho orientado a objetos;
 - desenho de interfaces de usuário;
 - realização de testes;
 - gerência da qualidade
 - gerência de projetos
- aplicar os padrões acima em projetos de software de pequeno e médio porte;
- participar de revisões técnicas, relatá-las e conduzi-las.

2 Ementa

Conceitos básicos. Processos de desenvolvimento de software. Introdução à UML. Engenharia dos requisitos de software. Análise e detalhamento dos requisitos de software. Revisões. Desenho. Teste. Gestão de Projetos. Gestão da Qualidade.

3 Programa detalhado

1. Engenharia de software: conceitos básicos. Produtos de software: requisitos, prazos, custos e qualidade.
2. Processos em engenharia de software. Elementos constituintes de um processo. Exemplos de processos. Estudo detalhado de um processo de desenvolvimento de software.
3. Introdução a UML. Modelagem. Visão geral: Coisas, Relacionamentos e Diagramas, Regras, Mecanismos Comuns. Classes. Relacionamentos, Colaborações.
4. Engenharia dos requisitos de software. Padrão para propostas de especificação. Padrão para especificação de requisitos. Principais técnicas de captura de requisitos: uso da prototipagem, JAD e relacionamento com clientes.
5. Análise de software. Técnicas de análise orientada a objetos: identificação de classes, relacionamentos, operações e atributos. Validação dos modelos.
6. Revisões de software. Princípios, preparação, condução e relato das revisões técnicas. Variantes de revisões: revisões individuais, revisões gerenciais, revisões de apresentação e inspeções. Papel das revisões técnicas no controle da qualidade.
7. Desenho de software. Uso do desenho orientado a objetos. Modularização do desenho: critérios para partição, visibilidade e navegação. Classes para estruturas de dados. Padrão para documentação do desenho. Desenho da interface de usuários. Diretrizes básicas para interfaces de usuário. Estilos de interação.

8. Testes de software. Princípios dos testes: objetivos e métodos. Tipos de teste: de integração, de aceitação e de regressão. Desenho de testes. Realização de testes.. Padrão para documentação de testes
9. Gestão de projetos. Gestão de requisitos. Planejamento de projetos. Técnicas para estimativa de projetos de software. Dimensionamento do tamanho, esforço e custos.
10. Gestão da Qualidade. Garantia da Qualidade. Gestão de Revisões. Gestão de configurações.

4 Trabalho Prático

O assunto de que trata a Engenharia de Software dificilmente é aprendido sem a realização de projetos práticos. Ao longo do semestre os alunos estarão envolvidos no desenvolvimento de um projeto de software para gestão de uma locadora de vídeos. Como cenário, deve-se supor que se trata de um produto de prateleira, isto é, um produto desenvolvido para colocação no mercado (segundo a norma [IEEE 1062], COTS - Commercial-of-the-shelf, é um produto estável e é normalmente bem definido em termos da documentação e possui funcionalidades e limitações conhecidas). Deseja-se a criação de um produto de boa qualidade, que facilite a administração da vídeo-locadora e estimule seu relacionamento com seus clientes. A especificação detalhada do produto cabe aos alunos, OBSERVANDO AS SEGUINTE DIRETRIZES.

- O trabalho deve ser realizado por grupos de 3 a 4 alunos;
- o produto poderá contemplar as necessidades internas de administração da vídeo-locadora mas deverá também atender ao relacionamento com o cliente da oficina;
 - A vídeo locadora deverá prover acesso ao programa a seus clientes, mas esse acesso não necessita ser simultâneo com o acesso por funcionários.
- o produto vai ser desenvolvido durante o semestre letivo, portanto, não há tempo para o desenvolvimento de um software muito complexo;
- por outro lado, um produto muito simplificado não ilustra adequadamente as questões de desenvolvimento de software que são objeto da Disciplina; portanto, cabe aos alunos, com bom senso, dimensionar adequadamente a complexidade de seu trabalho;
- o software deverá ter dimensão adequada para que se possa elaborar a especificação dos requisitos, o modelo de análise, o modelo de desenho (parcial), o desenho dos testes, a memória de planejamento e uma implementação parcial até o fim do semestre letivo;
- parte do trabalho envolvendo a especificação de requisitos não funcionais será feita no contexto de um trabalho de pesquisa que está sendo desenvolvido pelo professor e um aluno de mestrado. Este trabalho será detalhado a parte.
 - Os alunos da disciplina são solicitados a colaborar neste trabalho que lhes será, também, enriquecedor em termos de conhecimento adquirido.
- os grupos de trabalho serão indicados pelo professor;
- não serão admitidas cópias, mesmo que parcial, de trabalhos não realizados pelos alunos de cada grupo. Os exemplos apresentados no sítio do professor são apenas ilustrativos e, muitas vezes, trabalhos de alunos de turmas anteriores. Também não serão aceitas cópias de partes dos exemplos nos trabalhos;
- os alunos terão acesso a exemplos e modelos que podem ser usados para auxílio em seus trabalhos;
- trabalhos copiados de colegas ou ex-alunos ou de exemplos do Mercê serão severamente penalizados. Os exemplos apresentados no site do autor do livro texto são apenas ilustrativos, não devem ser simplesmente copiados nos trabalhos;
- os alunos deverão fazer registros de todo o esforço (número de horas) dedicado à realização do trabalho;
- para cada grupo de alunos realizando um trabalho, será indicado um grupo parceiro. Os participantes do grupo parceiro farão o papel de desenvolvedores pares dos autores nos trabalhos de revisão durante o desenvolvimento do software;
- os artefatos produzidos nos trabalhos deverão ser entregues impressos na recepção do DCC, assinando comprovante da entrega. Também deverá ser entregue uma versão eletrônica dos documentos, no sítio Web do professor (no endereço <http://homepages.dcc.ufmg.br/~clarindo/trabalho-engenharia-software> onde haverá uma pasta para cada grupo de alunos).

- A Especificação de Requisitos, incluindo o modelo de análise, deverá, antes de ser entregue, ser submetida a uma Revisão Técnica com a participação do grupo parceiro e resultado da revisão técnica também será entregue e avaliado;
- todos os trabalhos entregues deverão estar bem documentados conforme estabelecido na metodologia Praxis e a conformidade com os padrões também será avaliada. O resultado de revisões técnicas também deve ser bem documentado, incluindo o relatório da revisão e os comentários feitos pelos revisores com a identificação dos autores;
- o trabalho deverá ser entregues no prazo estabelecido. É importante o cumprimento do planejamento da disciplina, com os trabalhos sendo entregues na data certa para evitar desencontros. Por este motivo, serão descontados 10% dos pontos de cada trabalho por dia de atraso na entrega, sendo que na última entrega de trabalho não será admitido atraso;
- ao avaliar um trabalho, será feita uma verificação com uma "check-list" de conformidade com o padrão estabelecido pelo Praxis. Trabalhos incompletos serão penalizados;
- no trabalho deverão ser definidos pelo menos 8 casos de uso não triviais. Isso é uma diretriz, não se deseja uma atitude dos alunos de "inventar" casos de uso até chegar a esses números; eles devem aparecer normalmente como resultado do trabalho de desenvolvimento. Os fluxos de requisito, análise e gestão de projetos e de qualidade deverão contemplar todos os casos de uso, o fluxo de desenho deverá contemplar todos os casos de uso para desenho de interfaces do usuário e de dados persistentes e detalhar os fluxos de caso de uso e a realização de pelo menos um caso de uso, e os fluxos de implementação e teste deverão contemplar pelo menos um caso de uso não trivial;
- o tempo previsto nas aulas de laboratório é suficiente apenas para que um grupo apresente seu trabalho ou se inicie um trabalho como um exemplo ilustrativo de como funciona de determinada técnica da metodologia;
- como parte dos mecanismos de garantia da qualidade, usaremos também revisão técnica, após ser ensinado os mecanismos de revisões, em que o documento de Especificação de Requisitos, incluindo o modelo de Análise, será revistos pelos próprios colegas, por grupos parceiros. Cada grupo deverá entregar o relatório de revisão completo, incluindo a parte onde houve a participação do grupo parceiro. O grupo parceiro vai atuar no papel revisor, que normalmente é exercido por desenvolvedores de equipes que não estejam atuando diretamente naquele projeto. O grupo parceiro de cada grupo desenvolvedor será definido pelo professor. No trabalho entregue pelos alunos, o relatório de revisão resultará em duas notas: uma nota será dada ao grupo autor da revisão e a outra nota será dada ao grupo parceiro pela sua participação na revisão;
- entrega parcelada ao longo do semestre, conforme tabela abaixo, dividido em 4 partes.

A Tabela abaixo define a documentação (artefatos) a ser entregue em cada etapa do trabalho e o mostra o relacionamento entre os trabalhos propostos e o material suplementar fornecido. Este material compreende arquivos com exemplos e com modelos para preenchimento. O uso do modelo pode facilitar bastante o preenchimento dos documentos conforme os padrões. Os exemplos de modelos de análise e desenho são fornecidos em arquivos do Rational Rose. Os demais são arquivos do Microsoft Word.

Etapas	Resultado	Aula de entrega	Arquivo de exemplo	Arquivo de modelo
1	Proposta de Especificação	8	Merci_10_PESw	PESw
2	Especificação de Requisitos excluindo a seção 4	14	Merci_10_ERSw	ERSw;
3	Documentação do modelo de análise (seção 4 da Especificação de Requisitos)	17		ERSw (seção 4)
4	Relatório de revisão da ER	20	Merci_10_RRERSw	RRERSw
5	Memória de Planejamento do	28	Merci_10_MPPSw	MPPSw; PDSw,

	Projeto; plano de desenvolvimento do software; plano da qualidade do software		Merci_10_PDSw Merci_10_PQSw	PQSw
6	Descrição do Desenho, Descrição de testes; Relatório de testes	30	Merci_10_DTSw Merci_10_RTSw	DTSw, RTSw

Tabela - Planejamento da entrega do trabalho**5 Prova**

Além do trabalho prático, haverá duas provas. As provas serão sobre a matéria completa da disciplina até a data da prova.

Proposta de especificação de software	4
Especificação e análise de requisitos revisadas	14
Relatório da revisão da especificação e análise de requisitos do grupo autor	3
Revisão da especificação e análise de requisitos do trabalho do grupo parceiro	3
Memória de planejamento de projeto	6
Descrição de testes e relatório de testes	8
Plano de desenvolvimento do software	3
Plano da qualidade do software	3
Descrição de desenho	3
Prova	50
Avaliação em sala de aula e assiduidade (máximo)	7
Total	100

6 Comunicação

Os alunos poderão acessar o sítio do professor em <http://www2.dcc.ufmg.br/~clarindo/disciplinas/es/index.htm> para a obtenção de informação e material didático da disciplina. Também o sítio do autor do livro em <http://www.wppf.uaivip.com.br/> deve ser utilizado para o acesso a material didático.

O professor pode ser consultado através do e-mail : clarindo@dcc.ufmg.br. Neste caso, o campo de “assunto” da mensagem deve começar com “Engenharia de Software:”, seguido do assunto. Por exemplo: “Engenharia de Software: esclarecimento sobre o trabalho”.

7 Bibliografia

7.1 Livro Texto

Paula F., W. P. Engenharia de Software: Fundamentos, Métodos e Padrões, 2ª edição, LTC, 2003

7.2 Referências principais

- Fowler, M. "UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd Edition, Addison-Wesley, 2003
- [FOW 00] Fowler, M. "UML Essencial: um breve guia para linguagem-padrão de modelagem de objetos". Trad. Vera Pezerico e Christian Thomas Price. 2. Edição, Bookman, 2000
- Booch G. et al, "The Unified Modeling Language User Guide", Addison Wesley, 1999
- Jacobson I. et al, "The Unified Software Development Process", Addison Wesley, 1999
- Pressman, R.S. Engenharia de Software, Makron, 1995.
- Royce, W. "Software Project Management A Unified Framework", Addison Wesley, 1998
- Jones, T. C. "Estimating Software Costs", McGraw Hill, 1998
- Rumbaugh J. et al, "The Unified Modeling Language Reference Manual", Addison Wesley, 1999
- Conallen, J. "Building Web Applications With UML", Addison Wesley, 1999
- Humphrey, W. S. "Introduction to the Personal Software Process" Addison Wesley, 1997
- Penker, M.; Hans-Erik, E. Business Modeling with UML: Business Patterns at Work, Wiley, 2000
- Jones, P. H. "Handbook of Team Design" McGraw Hill, 1997
- Robertson, S.; Robertson, J. "Mastering the Requirements Process" Addison Wesley, 1999
- Constantine, L.L.; Lockwood, L.D. "Software for Use", Addison Wesley, 1999
- McConnell, S. "Software Project Survival Guide" Microsoft Press, 1997
- Yourdon, E. "Death March The Complete Software Developer's Guide to Surviving "Mission Impossible" Projects", Prentice Hall PTR
- DeMarco, Tom, "The Deadline A Novel About Project Management", Dorset House, 1998
- Rumbaugh J. et al, "Modelagem e Projetos Baseados em Objetos", Editora Campus, 1994
- Ghezzi C. et al, "Fundamentals of Software Engineering", Prentice-Hall International, 1991
- Myers G. J., "The Art of Software Testing", Wiley Interscience, 1979
- Furlan J. D., "Modelagem de Objetos Através da UML", Makron Books, 1998
- NIELSEN, J. *Usability Engineering*. Chestnut Hill, MA, Academic Press, 1993.
- Fowler M., "UML Distilled – Applying the Standard Object Modeling Language", Addison Wesley, 1997
- McCarthy, J. "Dynamics of Software Development" Microsoft Press, 1995
- IEEE 1062 *Recommended Practice for Software Acquisition*. The Institute of Electrical and Electronics Engineers, 1998.

8 Planejamento de aulas

Aula	Data	Assunto	Entrega do trabalho
1	7/8	Introdução à Engenharia de Software	
2	9/8	Processos de desenvolvimento de software	
3	16/8	Processos de desenvolvimento de software	
4	21/8	Desenho do Praxis	
5	23/8	Requisitos: conceitos, qualidade da especificação	
6	28/8	Requisitos: Padrão de propostas e de ER	
7	30/8	Requisitos: fluxo de requisitos; casos de uso, atores	
8	4/9	Requisitos: fluxo de requisitos; casos de uso, atores	Entrega Parte 1 do trabalho
9	6/9	Requisitos: Organização de casos de uso, qualidade	
10	11/9	Requisitos: prototipagem; relacionamento com clientes;	
11	13/9	UML – Visão Geral –classes; relacionamentos	
12	18/9	Laboratório: JAD de Especificação de Requisitos	Grupos: 3 e 4
13	20/9	Laboratório: JAD de Especificação de Requisitos	
14	25/9	UML : mecanismos comuns; diagrama de caso de uso; diagramas de classe e de objetos.	Entrega Parte 2 do trabalho
15	27/9	Análise: identificação de classes e relacionamentos.	
16	2/10	Análise: identificação de classes e relacionamentos	
17	4/10	Revisão técnica	
18	9/10	Análise: identificação de operações e atributos	
19	11/10	Laboratório: revisão Técnica da Especificação de Requisitos	Grupos: 5 e 6
20	16/10	Gestão da qualidade . Gestão de Projetos: gestão de requisitos e planejamento.	
21	18/10	Gestão de Projetos: planejamento	Entrega Parte 3 do trabalho.
22	23/10	Gestão de Projetos: estimativa de tamanho	
23	25/10	Prova. Caps.: 1, 2, 5, 6, 11.1 a 11.3, 13, 14, 15, 22	
24	30/10	Gestão de Projetos: controle	Entrega Parte 4 do trabalho
25	1/11	Desenho: fluxo; desenho arquitetônico, detalhamento	
26	6/11	Desenho: detalhamento, documentação	
27	8/11	Desenho de interfaces de usuário: conceitos básicos e diretri-	
28	13/11	Testes: princípios, tipos e atividades; testes de integração.	Entrega Parte 5 do trabalho
29	27/11	Testes: testes de aceitação e de regressão; documentação de	
30	29/11	Prova: Caps.: 1, 2, 5, 6, 11.1 a 11.3, 13, 14, 15, 22 7.1 a 7.3.1, 8, 10.1 a 10.4, 17.1 a 17.2, 18.1 a 18.3, 21	Entrega Parte 6 do trabalho

