

CURSO:	Engenharia de Software	SEMESTRE/ANO:	01/2017
DISCIPLINA:	Métodos de Desenvolvimento de Software	CRÉDITOS:	04
CARGA HORÁRIA:	60 horas	TURMA:	A
PROFESSOR:	Carla Silva Rocha Aguiar		

PLANO DE ENSINO

1 Objetivos da Disciplina

Métodos de desenvolvimento de software podem ser entendidos como conjuntos estruturados de boas práticas, podendo ser repetíveis durante o processo produtivo de software.

Diferentes métodos podem ser combinados e organizados em um modelo que defina o ciclo de vida de desenvolvimento, ou seja, as fases necessárias para o desenvolvimento de um projeto de software.

Nesse contexto, a disciplina *Métodos de Processos de Software* se faz importante para os futuros Engenheiros de Software por apresentar diferentes métodos de desenvolvimento, com enfoque especial aos diferentes ciclos de vida e técnicas de desenvolvimento de software. Os principais objetivos são:

- Capacitar o aluno a compreender os diferentes métodos, ferramentas, procedimentos e paradigmas de desenvolvimento de software.
- Capacitar o aluno a aplicar / adaptar processos de desenvolvimento de software à resolução de problemas de software.

2 Ementa do Programa

Modelos de ciclo de vida e de processos; Processo Unificado. Métodos Ágeis de desenvolvimento de software. Outras abordagens de desenvolvimento de software (orientado a dados, orientado a funções, orientado a objetos, orientado a aspectos). Ferramentas.

3 Descrição do Programa

I. *Processos de Desenvolvimento de Software*

i. Modelos de Processo de Desenvolvimento

de Software (ciclo de vida)

ii. Atividades de Processo

II. *Fundamentos do Processo Unificado de Desenvolvimento de Software*

- i. Conceitos
- ii. Fases
 - Iniciação, Elaboração, Construção e Transição
- iii. Disciplinas
 - Modelagem de Negócio, Requisitos, Análise e Desenho, Implementação, Teste, Gerenciamento de Projeto, Gerência de Configuração e Mudanças, Implantação e Ambiente

III. *Fundamentos do Extreme Programming*

- i. O manifesto Ágil
- ii. Os Quatro valores e as Quatro variáveis
- iii. O jogo do planejamento

iv. Releases Pequenas

- v. A metáfora
- vi. Histórias do Usuário
- vii. Desenho simples
- viii. Testes (unitário, aceitação)
- ix. Refatoração
- x. Programação em Pares
- xi. Desenvolvimento Coletivo

IV. *Outras Abordagens/Paradigmas de Desenvolvimento de Software*

- i. Método orientados a funções e dados
- ii. Método Orientado a Aspectos
- iii. Método Orientado a Serviços
- iv. Método Orientado a Modelo

4 Horário das aulas e atendimento

AULAS TEÓRICAS: terças e quintas, das 14:00 às 15:50 hrs.

AVALIAÇÕES: Mini-Testes, Projeto; e Participação.

5 Dinâmica das Aulas

A carga horária da disciplina é de 60 horas. Esta carga horária é realizada em encontros presenciais das 14:00 às 15:50 hs, duas vezes por semana. Ao longo do semestre, alguns destes encontros poderão ser substituídos por aulas via moodle a ser previamente agendado pelo Prof. responsável.

Metodologia

O método aplicado considerará o desenvolvimento de habilidades de escrita, leitura/compreensão, apresentação oral, trabalho em grupo e produção no contexto de metodologias de Engenharia de Software. O material de apoio está disponível no moodle, de forma que o aluno possa se planejar em relação a aquisição do conhecimento da disciplina. Durante as aulas poderão acontecer atividades assim organizadas:

Atividade Intra-Classe
Aula expositiva
Produção de Software (aula prática)
Apresentação
Arguição do Professor

- A aula expositiva contemplará aqueles assuntos considerados pelo professor como os mais importantes de serem destacados. Além da consideração do professor, as dúvidas que emergirem durante a atividade de Produção de Software também serão abarcadas na aula expositiva.
- A Produção de Software se dará em relação ao projeto da disciplina, que também será desenvolvido nas atividades intra-classes, de forma colaborativa com o auxílio do professor. Como produto dessa atividade serão elaborados artefatos relacionados as metodologias que serão trabalhadas na disciplina.
- Já em relação a participação, cada aluno deverá apresentar o resultado do seu trabalho dentro de sala de aula quando da apresentação do grupo. No momento da avaliação serão considerados: organização do tempo, postura, domínio do assunto, corretude, clareza de objetivos, além dos aspectos técnicos do desenvolvimento.

Para o desenvolvimento das atividades supracitadas ocorrerá a utilização do laboratório de informática para o uso de softwares aplicativos. Talvez não tenhamos laboratório totalmente operacional nesse semestre.

A fim de fortalecer a aprendizagem da disciplina, as aulas serão complementadas com atividades de projeto em demandas extra-classe. Estas atividades serão desenvolvidas com acompanhamento do professor, bem como através da Plataforma Moodle; ou do grupo (privado) da disciplina no Facebook, além da wiki da disciplina.

6 Avaliações e Critérios de Avaliação

A avaliação será feita por meio de:

- EP1 a EPn: Entregas do Projeto.
- MT1 a MTn: Mini-Teste Individual e Presencial (Prova).
- P1: Participação em sala de aula

Para avaliar o desempenho nas aulas teóricas, serão aplicados mini-testes ao longo do semestre (número a ser definido). A avaliação será individual e sem consulta a qualquer material didático. A duração de cada mini-teste será de 15 minutos, e será realizado no início da aula. A aplicação dos mini-testes não será previamente comunicada. Portanto, é importante manter os estudos em dia!

O objetivo do Projeto é integrar as disciplinas Gestão de Projetos e Portfólio (GPP) e Métodos de Desenvolvimento de Software (MDS), de forma que os alunos façam um trabalho complementar, simulando uma situação real de desenvolvimento de software e gestão do projeto. Enquanto os alunos de MDS irão se concentrar na execução metodologia de desenvolvimento através da especificação de requisitos, codificação e testes,

os alunos de GPP irão se concentrar na gestão de tarefas, distribuição de recursos, estimativas do projeto, garantindo assim a correta aplicação da metodologia de desenvolvimento. Haverá duas avaliações formais das releases a serem desenvolvidas.

Os pesos atribuídos aos diferentes eventos de avaliação são indicados abaixo.

Evento de Avaliação	Peso
Mini-Teste	40%
Projeto	50%
Participação	10%

Para o cálculo da média final da disciplina, consideram-se as seguintes medidas:

- Média das entregas do Projeto:

$$\overline{MEP} = Nota1 * 0.40 + Nota2 * 0.6 \quad (1)$$

onde

$$Nota1 = \frac{\sum_{i=1}^n Nota1_i * p_i}{\sum_{i=1}^n p_i} \quad (2)$$

$$Nota2 = \frac{\sum_{i=1}^n Nota2_i * p_i}{\sum_{i=1}^n p_i} \quad (3)$$

Nota1 = Entregas RUP e Notas2 = Entregas XP

- Média mini-testes:

$$MMT = \frac{\sum_{i=1}^n MT_i}{n} \quad (4)$$

- Média de Participação:

$$MP = \frac{\sum_{i=1}^n P_i}{n} \quad (5)$$

- **Média final:**

$$MF = (MMT * 0.4) + (MEP * 0.5) + (MP * 0.1) \quad (6)$$

Para o aluno satisfazer os seguintes requisitos para obter a aprovação na disciplina:

- Aprovação se $MF \geq 5,0$ e se Percentual de faltas (PF) for $PF \leq 25\%$. Onde PF é dado pelo número de aulas com faltas registradas dividido pelo número de aulas ministradas.
- Reprovação se $MF < 5,0$ ou se $PF > 25\%$. Nessa situação o aluno será considerado reprovado por nota ou por falta.

6.1 Avisos

1. Pontualidade na entrega das avaliações - o atraso será penalizado. Perde-se 5% do valor da nota dada a cada entrega do projeto para cada dia de atraso na entrega. Todas as entregas dos projetos deverão ser postados no Moodle em datas previamente definidas pelo professor responsável.
2. Também são considerados critérios de avaliação da participação: assiduidade; pontualidade; interesse; participação em sala. A nota de participação é de autonomia do professor.
3. Os documentos referentes à disciplina estarão disponíveis no Moodle.
4. Os casos não previstos de perda de avaliação serão avaliados individualmente, de acordo com as circunstâncias.
5. O uso do aparelho celular durante o período da atividade em sala de aula não é permitido.
6. As avaliações compõe 50% das notas das provas. O restante da nota individual do aluno será feita pela sua contribuição no wiki da disciplina. Espera-se que o aluno tenha, pelo menos, duas contribuições por semana. Essa contribuição é individual. O que será avaliado é a corretude da contribuição, qualidade, e a relevância da contribuição.
7. O aluno que perder uma avaliação poderá fazer uma avaliação de reposição, caso a justificativa da ausência seja motivo de saúde. Neste caso o aluno deverá comprovar tal situação por meio de atestado médico entregue ao professor no seu retorno às atividades. O teste de reposição poderá ser aplicado também em outros casos amparados legalmente
8. Os mini-testes não serão agendados previamente. Portanto, mantenham o conteúdo da disciplina sempre up to date! Na composição da média haverá o descarte da menor nota.
9. As entregas intermediárias do projeto totalizam 10% da Nota 1 e 2
10. A arquitetura deve contemplar no mínimo o padrão de projeto MVC e o tratamento de exceções
11. A cobertura de código deverá ser 90%, excetuando a camada de apresentação

7 Cronograma

Semana	Aula	Data	Conteúdo
01	1	07/03	<i>Apresentação do plano de ensino disciplina + Apresentação sobre o trabalho da disciplina</i>
	2	09/03	<i>Fundamentos do Processo Unificado no contexto do IBM/Rational Unifed Process(RUP) - O RUP no contexto das boas práticas</i>
02	3	14/03	<i>Fundamentos do Processo Unificado no contexto do IBM/Rational Unifed Process (RUP)- A organização das fases no RUP</i>
	4	16/03	<i>Fundamentos do Processo Unificado no contexto do IBM/Rational Unifed Process (RUP) - Disciplinas I: (I) Modelagem de Negócios, (II) Ambiente</i>
03	5	21/03	<i>Fundamentos do Processo Unificado no contexto do IBM/Rational Unifed Process (RUP) - Disciplinas I: (I) Gerenciamento de Projeto, (II) Requisitos</i>
	6	23/03	<i>Fundamentos do Processo Unificado no contexto do IBM/Rational Unifed Process (RUP) - Disciplinas II: (I) Análise & Design, (II) UML Nota1/Entrega (Etapa/Feedback 1): (I) Documento de Visão do Projeto, (II) Planejamento/Comunicação Interna e Externa (agenda de trabalho + ferramentas)</i>
04	7	28/03	<i>Fundamentos do Processo Unificado no contexto do IBM/Rational Unifed Process (RUP) - Disciplinas II: (I) Análise & Design, (II) UML</i>
	8	30/03	<i>Fundamentos do Processo Unificado no contexto do IBM/Rational Unifed Process (RUP) - Disciplinas II: (I) Implementação, (II) Teste</i>
05	9	04/04	<i>Fundamentos do Processo Unificado no contexto do IBM/Rational Unifed Process (RUP) - Disciplinas II: (I) Implementação, (II) Gerenciamento de Configuração e Mudança Nota1/Entrega (Etapa/Feedback 2):(I) Documento de Arquitetura do Projeto, (II) Especificação dos casos de uso, (III) Especificação dos casos de teste, (IV) Configuração do Repositório (Github), (V) Wiki atualizada (VI) Protótipo de alta fidelidade</i>
	10	06/04	<i>Introdução a MDS - Modelos de Ciclos de Vida de Desenvolvimento de Software</i>
06	11	11/04	<i>Modelos de Ciclos de Vida de Desenvolvimento de Software</i>
	-	13/04	Permuta da aula
07	-	18/04	Permuta da aula
	12	20/04	<i>Apresentação dos Trabalhos - Release 1 Nota1/Entrega (Etapa/Feedback 3 (40%)): (I) Apresentação das informações do Documento de Visão (10%), (II) Entrega/Apresentação da especificação de casos de uso e casos de teste + protótipo de alta fidelidade (10%), (III) Relatório da execução dos testes (10%), (IV) Entrega/Apresentação da 1a. Release do Projeto (arquitetura) (60%)</i>
08	13	25/04	<i>Dinâmica Motivacional relacionada a formação de times ágeis</i>
	14	27/04	<i>Fundamentos do eXtremeProgramming (XP):(I) O manifesto Ágil,(II) Quatro valores, (III) Quatro variáveis</i>

Semana	Aula	Data	Conteúdo
09	15	02/05	Fundamentos do eXtremeProgramming (XP): Histórias do Usuário Nota2/Entrega (Etapa/Feedback 1): (I) Backlog de produto e backlog de sprint, (II) Repositório/Commits frequentes/Testes Automatizados (III) Integração contínua
	16	04/05	Fundamentos do eXtremeProgramming (XP): (I) Visão Geral sobre o Behavior Driven Development (BDD), (II) O jogo do planejamento
10	17	09/05	Fundamentos do eXtremeProgramming (XP): (I) Releases Pequenas, (II) A metáfora
	18	11/05	Fundamentos do eXtremeProgramming (XP): Testes (unitário, integração, aceitação) Nota2/Entrega (Etapa/Feedback 2): (I) Wiki atualizada, (II) Histórias de usuário (com critérios aceitação) + Testes de Aceitação Automatizado
11	19	16/05	Fundamentos do eXtremeProgramming (XP): Refatoração
	20	18/05	Fundamentos do eXtremeProgramming (XP): (I) Cobertura de Código, (II) Integração contínua
12	21	23/05	Fundamentos do eXtremeProgramming (XP): (I) Semana de 40-hs, (II) Cliente na equipe, (III) Padrão de Codificação. Nota2/Entrega (Etapa/Feedback 3): (I) Tracking, (II) Repositório/Commits frequentes/Testes Automatizados (III) Integração contínua
	22	25/05	Fundamentos do eXtremeProgramming (XP): (I) Desenho simples, (II) Programação em Pares, (III) Desenvolvimento Coletivo
13	23	30/05	Desenvolvimento Ágil Nota2/Entrega (Etapa/Feedback 4): (I) Wiki atualizada, (II) Análise Métricas Qualidade (III) Análise Velocity
	24	01/06	Desenvolvimento Ágil
14	25	06/06	Desenvolvimento Ágil
	26	08/06	Desenvolvimento Ágil Nota2/Entrega (Etapa/Feedback 5): (I) Wiki atualizada, (II) Refatoração
15	27	13/06	Desenvolvimento Ágil
	28	15/06	Desenvolvimento Ágil
16	29	20/06	Desenvolvimento Ágil
	30	22/06	Apresentação dos Trabalhos - Release 2 - Nota2/Entrega Etapa/Feedback 2 - Encerramento do Projeto (Entrega Final). Considerar:) (I) Histórias e Planejamento das releases (15%), (II) Testes Automatizados e Cobertura de código + Testes de Aceitação automatizado (50%), (III) Tracking (15%), (IV) Wiki Atualizada (5%), (IV) Software Implantado e Disponibilizado para Uso (5%)
17	31	27/06	Apresentação dos Trabalhos - Release 2
	-	29/06	Entrega das menções finais. Revisão de notas
18	-	04/07	Fim do semestre

8 Bibliografia

- Bibliografia Basica

- Beck, K., Programacao Extrema (XP) Explicada, 1st ed. Bookman, 2004.
- Jacobson, I., Booch G., Rumbauch J., The Unified Software Development Process, 1st ed., Addison-Wesley, 1999.

EBRARY Lano, K., UML 2 Semantics and Applications, 1st ed., Wiley, 2009.

- Bibliografia Complementar

- Sommerville, I., Engenharia de software. 8th ed., Pearson Addison Wesley, 2007.
- Pfleeger, S. L., Engenharia de software: teoria e pratica. 2nd ed., Prentice Hall, 2004.
- Pressman, R. S., Engenharia de software. 6th ed., McGraw-Hill, 2006.
- Ambler, S., Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process, 1st ed., Wiley, 2002
- Jacobson, I., Booch G., Rumbauch J., UML: Guia do Usuario, 2nd ed., Elsevier, 2005.

OPEN ACCESS Scrum e XP direto das Trincheiras. (<http://www.infoq.com/br/minibooks/scrum-xp-from-the-trenches>)