

CURSO:	Engenharia de Software	SEMESTRE/ANO:	01/2018
DISCIPLINA:	Engenharia de Produto de Software	CRÉDITOS:	04
CARGA HORÁRIA:	60 horas	TURMA:	A
PROFESSOR:	Carla Silva Rocha Aguiar		

PLANO DE ENSINO

1 Ementa do Programa

Teorias que suportam o Desenvolvimento de Produtos. Diferenças e semelhanças da Produção na Manufatura e no Software. Gestão do escopo, tempo, custos, qualidade, recursos materiais, pessoas, comunicações, riscos em diferentes paradigmas (ex: tradicional e ágil). Gestão estratégica de projetos (portfólios e programas). Gestão de Projetos de software no contexto das normas e dos modelos de melhoria de processo de software. Projetos e implementação de sistema usando modelos de gerenciamento.

2 Descrição do Programa

- | | |
|---|---|
| <p>I. <i>As origens do gerenciamento</i></p> <p> i. A teoria da administração científica</p> <p>II. <i>Gerenciamento de Projetos Tradicional</i></p> <p> i. Conceitos Básicos</p> <p> ii. Contextualização do gerenciamento de projetos</p> <p> iii. Ciclo de Vida de Projeto e Produto</p> <p> iv. Grupos de Processo</p> <p> v. Áreas de Conhecimento (escopo, custo, tempo,...)</p> <p> vi. Ferramentas</p> <p>III. <i>O pensamento sistêmico e a complexidade no contexto do gerenciamento</i></p> <p> i. Apresentação da teoria geral dos sistemas</p> <p> ii. Apresentação da teoria da complexidade</p> <p> iii. Apresentação da teoria das restrições</p> | <p> iv. Perfil do EP</p> <p> v. Estrutura</p> <p> vi. Medição</p> <p> vii. Ferramentas</p> <p>IV. <i>Gerenciamento Ágil de Projetos</i></p> <p> i. Conceitos Básicos</p> <p> ii. O manifesto Ágil</p> <p> iii. SCRUM</p> <ul style="list-style-type: none"> • Principais Papeis • Começando um projeto com Scrum • Ciclo de vida do Scrum • Planejando um Sprint • Product Backlog e Sprint Backlog • Daily-Meeting • Gerenciando o Sprint • Ferramentas <p>V. <i>Gestão Estratégica de Projetos</i></p> |
|---|---|

i. Portfólios e Programas

ii. Conceitos

iii. Escritório de Projetos-EP

VI. 06. *Projetos e implementação de sistema*

i. Produção de artefatos produzidos a cada

iteração do ciclo de desenvolvimento do projeto

ii. Avaliação da qualidade interna do produto de software

iii. Implantação do projeto em ambiente de produção

3 Horário das aulas e atendimento

AULAS TEÓRICAS: terças e quintas, das 10:00 às 11:50 hrs.

AVALIAÇÕES: Testes, Projeto; e Participação.

4 Dinâmica das Aulas

A carga horária da disciplina é de 60 horas. Esta carga horária é realizada em 2 encontros presenciais de 1:50 hs ,2 vezes por semana.

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 na organização Github da disciplina, de forma que o aluno possa se planejar em relação a aquisição do conhecimento da disciplina.

Durante as aulas haverá 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. Em princípio, cada grupo deverá apresentar o resultado de seu trabalho em três eventos, podendo se estender a 4 apresentações. No momento da avaliação serão considerados: organização do tempo, postura, domínio do assunto, corretude, clareza de objetivos.

A fim de fortalecer a aprendizagem da disciplina, as aulas serão complementadas com atividades de projeto em demandas extra-classe.

5 Avaliações e Critérios de Avaliação

A avaliação será feita por meio de:

- EP1 a EPn: Entregas do Projeto.
- MT1 a MTn: 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 (n0 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 Engenharia de Produto de Software (EPS) 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 e gerência de configuração, 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
Teste	30%
Projeto	60%
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.

5.1 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.
3. Os documentos referentes à disciplina estarão disponíveis no GITHUB.
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. 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

7. Os mini-testes não serão agendados previamente. Portanto, mantenham o conteúdo da disciplina sempre up to date! Na composição da média *MMT* haverá o descarte da menor nota.
8. As entregas intermediárias do projeto totalizam 10% da Nota 1 e 2
9. A cobertura de código deverá ser 90%, excetuando a camada de apresentação

6 Cronograma

Semana	Aula	Data	Conteúdo
01	1	06/03	<i>Apresentacao do plano de ensino disciplina</i>
	2	08/03	<i>Apresentacao sobre o trabalho da disciplina - Entrega: Wiki organizada (contatos, estrutura, ferramentas, agenda de trabalho), Versão 0.1 README, Licença, Open Source Guides</i>
02	3	13/03	<i>Entrega Continua - Papeis agil</i>
	4	15/03	<i>Entrega Continua - Papeis devops - Entrega Comunicacao definida (canais, frequencias, etc) - Disponibilizar no README + licenca, planejamento de ferramentas.</i>
03	5	20/03	<i>Entrega Continua - Planejamento</i>
	6	22/03	<i>Entrega Continua - Monitoramento - Entrega: Roadmap projeto, EAP, Repositorio organizado, metodologia disponibilizada</i>
04	7	27/03	<i>Entrega Continua - Monitoramento e Controle</i>
	8	29/03	<i>Entrega Continua - Pipeline de deploy continuo - Entrega: Indicadores monitorados, definicoes de papeis, definicao de pronto, definicao de produtividade</i>
05	9	03/04	<i>Entrega Continua - Pipeline de deploy continuo</i>
	10	05/04	<i>Entrega Continua - Metricas - Qualidade de codigo I - Entrega: Ambiente de desenvolvimento pronto, ambiente de homologacao planejado, gitflow registrado, plano de qualidade, integracao/deploy continuos e metricas configurados</i>
06	11	10/04	<i>Entrega Continua - Gitflow - Conceitos de GCS + Controle de versao</i>
	12	12/04	<i>Entrega Continua - Integracao Continua - Entregas: integracao continua, testes, ambiente de homologacao, risk burndown</i>
07	-	17/04	
	13	19/04	<i>Apresentacao dos Trabalhos - Release 1 Nota1/Entrega (Etapa (40%)):</i> (I) <i>Codigo/deciso</i> es (30%), (II) <i>cobertura testes, integracao continua, deploy continuo, metricas, quantidade de historias de usuarios implementadas, solucao proposta, testes de aceitacao, ambiente de desenvolvimento</i> (30%), (III) <i>Produtividade - Monitoramento e controle (Tracking) - Acompanhamento/Desempenho do Projeto (Considerar as fases e produtos de trabalhos planejados)</i> (30%), (IV) <i>Riscos</i> (10%)
08	14	24/04	<i>O pensamento sistemico e a complexidade do gerenciamento - Pareamento Remoto (assinatura de commits)</i>
	15	26/04	<i>Entrega Continua - Metricas Qualidade deCodigo II - Entrega I: relatorio de deploy continuo + produtividade</i>