

Instituto de Informática
Departamento de Informática Aplicada

Dados de identificação

Disciplina: TÓPICOS ESPECIAIS EM COMPUTAÇÃO VIII

Período Letivo: 2016/1

Período de Início de Validade: 2016/1

Professor Responsável pelo Plano de Ensino: KARIN BECKER

Sigla: INF01055

Créditos: 4

Carga Horária: 60

Súmula

Assuntos relacionados a inovações tecnológicas decorrentes de pesquisas recentes ou a aplicações específicas, de interesse a um grupo restrito ou tendo caráter de temporalidade, enfocando aspectos não abordados ou abordados superficialmente em disciplinas regulares.

Currículos

Currículos	Etapa Aconselhada	Natureza
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO		Eletiva

Objetivos

O desenvolvimento da disciplina busca dar ao aluno, ao final do semestre, condições de:

1. Compreender a contribuição dos métodos e práticas ágeis às questões críticas de Engenharia de Software;
2. Conhecer os principais métodos ágeis, seus valores, princípios, artefatos e práticas ;
3. Aplicar de forma crítica conceitos e práticas ágeis em contextos de engenharia de software, relacionando questões às quais de adequam, condições e restrições de aplicação, e valor que agregam.

Conteúdo Programático

Semana: 1

Título: Introdução a métodos ágeis: princípios e valores

Conteúdo: 1. Introdução a métodos ágeis

- 1.1. Conceitos e Princípios básicos da Engenharia de Software
- 1.2. Métodos prescritivos
- 1.3. Manifesto ágil e princípios ágeis
- 1.4. Visão geral dos métodos ágeis
- 1.5. Adoção de Práticas e de Métodos Ágeis

Semana: 2 a 3

Título: Desenvolvimento Enxuto de Software

Conteúdo: 2. Desenvolvimento Enxuto de Software

- 2.1. Origem: Manufatura enxuta
- 2.2. Princípios de desenvolvimento enxuto de software
- 2.3. Kanban

Semana: 4 a 6

Título: SCRUM

Conteúdo: 3. SCRUM

- 3.1. Princípios Gerais
- 3.2. Framework SCRUM
 - 3.2.1. Papéis
 - 3.2.2. Atividades
 - 3.2.3. Artefatos
- 3.3. Histórias e Backlogs
- 3.4. Planejamento de Produto
- 3.5. Sprints
 - 3.5.1. Objetivos e propriedades

- 3.5.2. Planejamento
- 3.5.3. Backlog de Sprint
- 3.5.4. Reuniões
- 3.5.5. Instrumentos Visuais
- 3.5.6. Estimativas e Velocidade

Semana:	7
Título:	Histórias de Usuários
Conteúdo:	4. Histórias de Usuários
	4.1. Requisitos: Revisão
	4.2. Propriedades
	4.3. Formatos de Descrição
	4.4. Processo de identificação, expressão e validação de requisitos como histórias
	4.5. Critérios de Aceitação
Semana:	8 a 9
Título:	Gestão de Projetos Ágeis
Conteúdo:	7. Gestão de Projetos Ágeis
	7.1. Planejamento
	7.2. Estimativas
	7.2. Controle e Monitoramento
	7.3. Indicadores visuais
	7.4. Gestão de Riscos
Semana:	10 a 14
Título:	XP
Conteúdo:	5. XP
	5.1. Valores
	5.2. Práticas
	5.2.1. Metáforas
	5.2.2. Cliente presente
	5.2.3. Versões pequenas
	5.2.4. Refatoramento
	5.2.5. Padrões de Codificação
	5.2.6. Integração contínua
	5.2.7. Programação aos pares
	5.2.8. Desenvolvimento orientado a testes
	5.2.9. Jogo do Planejamento
Semana:	15
Título:	Modelagem Ágil
Conteúdo:	6. Modelagem Ágil
	6.1. Valores e Princípios
	6.2. Práticas
	6.3. Modelagem ágil e processos de desenvolvimento de software

Metodologia

A disciplina é apresentada em aulas teórico-práticas, nas quais são combinadas a apresentação de conceitos e técnicas, com sua aplicação prática.

Além de aulas expositivas, serão recomendadas leituras para aprofundamento do conteúdo teórico e reflexão sobre suas características e valor agregado.

A fixação do conteúdo teórico dá-se através de análise e discussão de exemplos, de dinâmicas que permitem fixar o conteúdo de forma lúdica, da realização de exercícios ou estudos de caso selecionados e de trabalhos extra-classe.

É particularmente importante na elaboração das atividades práticas em geral o envolvimento dos alunos na forma de equipes,

simulando os conceitos de time de desenvolvimento.

O moodle será utilizado como arcabouço de ensino, concentrando os planos de aula, o material didático disponibilizado pelo professor, comunicação e tarefas.

As 60 horas previstas para atividades teóricas e práticas indicadas neste Plano de Ensino incluem 30 encontros de 100 minutos de duração (2 períodos de 50 minutos por encontro, 2 encontros por semana, durante 15 semanas), num total de 3.000 minutos, e mais 10 horas (600 minutos) de atividades autônomas, realizadas sem contato direto com o professor, correspondentes a exercícios e trabalhos extraclasse

Carga Horária

Teórica: 45

Prática: 15

Experiências de Aprendizagem

Os alunos desenvolverão exercícios de fixação para os conteúdos vistos, bem como dinâmicas que permitam compreender práticas e conceitos subjacentes aos métodos ágeis, e ao desenvolvimento de software em equipes.

Os alunos deverão praticar através de projetos de pequeno porte, elaborados de forma incremental, os métodos estudados. Com isso, experimentarão, na prática, as dificuldades e vantagens inerentes ao processo de desenvolvimento. Cada incremento tem foco em práticas específicas, como planejamento de ciclo, teste automatizados, integração contínua, desenvolvimento aos pares, retrospectivas e revisões, entre outros.

Os alunos devem compartilhar com os colegas seus resultados e estimular uma discussão através da apresentação de trabalhos.

Critérios de avaliação

* Provas : Serão realizadas 2 (duas) provas, que ocorrem conforme calendário da disciplina em datas pré-definidas. As provas envolvem o conteúdo de toda disciplina ministrado até a prova. A avaliação de cada prova é divulgada em até 3 semanas após realização da prova, ou 72 horas antes da prova de recuperação. Peso 6 sobre o total, a razão do peso 3 para cada prova.

* Trabalhos Práticos : Serão realizados trabalhos práticos, organizados como uma combinação de atividades em classe e extraclasse. O trabalho progredirá de forma incremental, onde a cada etapa serão agregados novos elementos. Estão previstas aulas para acompanhamento da realização de práticas ágeis, para resolução de dúvidas e para apresentação. Os trabalhos práticos serão realizados em grupo e seus desenvolvimentos (parciais e/ou final) enviados via moodle nas datas indicadas e/ou apresentados. O conjunto dos trabalhos práticos tem peso 3 sobre o total. Haverá apresentações (parciais ou final) do trabalho, e seu resultado final divulgado até 72 horas antes da prova de recuperação.

* Participação: incluir a avaliação sobre realização/participação de exercícios e dinâmicas propostos, aulas de laboratório, posicionamento quanto a conteúdo e dúvidas, qualidade de participação em aula e motivação durante o desenvolvimento dos trabalhos, e assiduidade do aluno. Peso 1 sobre o total.

* Formação do Conceito Final : O conceito final do aluno será atribuído levando-se em consideração as provas (ou trabalhos teóricos), trabalhos práticos e participação, nos pesos acima definidos.

A média ponderada das provas, trabalhos e avaliação do professor será convertida em conceito, mediante escala abaixo:

Nota $\geq 9,0$ = A

Nota $\geq 7,5$ e $< 9,0$ = B

Nota $\geq 6,0$ e $< 7,5$ = C

Nota $< 6,0$ = D

Observações: Somente será calculada a média geral daqueles alunos que tiverem obtido um índice de frequência às aulas igual ou superior a 75% das aulas previstas. Aos que não satisfizerem este requisito, será atribuído o conceito FF (Falta de Frequência).

Atividades de Recuperação Previstas

O aluno que obtiver conceito final D pode realizar uma prova de recuperação versando sobre todo o conteúdo da disciplina. Se a nota obtida nessa prova for igual ou superior a 6,0, o conceito mudará para C.

Observações: Para poder realizar a prova de recuperação, o aluno deve ter realizado ao menos uma das provas/trabalhos teóricos E ter entregue o trabalho prático. Os que não se enquadrarem nesta situação permanecerão com conceito D.

Bibliografia

Básica Essencial

Cohn, M.. Succeeding with Agile: Software Development Using Scrum. Addison-Wesley, 2010. ISBN 978-0-321-57936-2.

Kent Beck.. Extreme programming explained : embrace change. Addison-Wesley, 2005. ISBN 9788536303871.

Poppendieck, M.. Lean software development: an agile toolkit.. Pearson, 2003. ISBN 0321150783.

Básica

Ambler, S.. Modelagem Ágil.. Bookman, 2004. ISBN 9788577804139.

Cohn, M.. User Stories Applied: For Agile Software Development. Pearson, 2004. ISBN 0-321-20568- 5.

Kenneth Rubin. Essential Scrum. Addison-Wesley, 2012. ISBN 9780137043293.

Lisa Crispin, Janet Gregory. Agile Testing: A Practical Guide for Testers and Agile Teams. Addison Wesley, 2009. ISBN 0321534468.

Mike Cohn.. Agile Estimating and Planning. Prentice Hall, 2005. ISBN 0131479415.

Complementar

Esther Derby, Diane Larsen. Agile Retrospectives: Making Good Teams Great. Pragmatic Bookshelf,, 2006. ISBN 0977616649.

Outras Referências

Não existem outras referências para este plano de ensino.

Observações

- Disciplina recomendada apenas aos alunos que já cursaram a disciplina Engenharia de Software N (INF01127).

- Conteúdo pode servir como embasamento teórico para a disciplina Laboratório de Engenharia de Software.