



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Curitiba



Informações da disciplina

Código Ofertado	Disciplina/Unidade Curricular	Modo de Avaliação	Modalidade da disciplina	Oferta
CSE42	Metodologias Ágeis Para O Desenvolvimento De Software	Nota/Conceito E Frequência	Presencial	Semestral

Carga Horária					
AT	AP	APS	ANP	APCC	Total
0	3	3	0	0	45
• AT: Atividades Teóricas (aulas semanais). • AP: Atividades Práticas (aulas semanais). • ANP: Atividades não presenciais (horas no período). • APS: Atividades Práticas Supervisionadas (aulas no período). • APCC: Atividades Práticas como Componente Curricular (aulas no período, esta carga horária está incluída em AP e AT). • Total: carga horária total da disciplina em horas.					

Objetivo		
Ao final da disciplina o aluno deverá ser capaz de: - Descrever e comparar alguns dos principais métodos e práticas ágeis - Determinar a adequabilidade dos métodos e práticas ágeis para um projeto em particular - Avaliar quão bem um projeto está seguindo os princípios ágeis, e auxiliar o projeto a tornar-se mais ágil (quando apropriado) - Entender o relacionamento entre o cliente e a equipe de desenvolvimento em projetos ágeis e as responsabilidades de ambas as comunidades - Conhecer parte da literatura científica e não-científica sobre métodos ágeis		
Ementa		
Introdução às metodologias ágeis. Software ágil: Valores e princípios fundamentais. Principais práticas das metodologias ágeis. Exemplos de metodologias ágeis.		
Conteúdo Programático		
Ordem	Ementa	Conteúdo
1	Introdução às metodologias ágeis. Software ágil: Valores e princípios fundamentais.	Surgimento dos Métodos Ágeis: contexto e motivações. Manifesto Ágil: o documento, seu significado e seu impacto. Valores e princípios ágeis: significado e impacto.

Ordem	Ementa	Conteúdo
2	Principais práticas das metodologias ágeis	Programação Pareada, Testes de Unidades, Desenvolvimento Dirigido por Testes (Test Driven Development TDD), Retrospectivas, Refatoração, Coding Dojos.
3	Exemplos de metodologias ágeis.	Kanban, Scrum, Programação Extrema (Extreme Programming), Lean Software Development

Bibliografia Básica
COCKBURN, Alistair. Agile software development : [b the cooperative game] . 2nd ed. Upper Saddle River, NJ: Addison-Wesley, c2007. xxxiv, 467 p. (The Agile software development series) ISBN 9780321482754.
POPPENDIECK, Mary; POPPENDIECK, Thomas David. Lean software development: an agile toolkist . Boston: Addison Wesley, c2003. xii, 203 p. (The agile software development series) ISBN 9780321150783.
SHORE, James; WARDEN, Shane. The art of agile development. Beijing; Sebastopol, CA: O'Reilly, c2008. xix, 415 p. (Theory in practice) ISBN 9780596527679.
BECK, Kent et al. Manifesto para o desenvolvimento ágil de software. Disponível em: http://manifestoagil.com.br/ . 2001.
KNIBERG, H.; SKARIN, M. Kanban e Scrum: obtendo o melhor de ambos. InfoQ, 2009. Disponível em: http://www.infoq.com/br/minibooks/kanban-scrum-minibook . Acesso em: 9 de agosto de 2016.
KNIBERG, H. Scrum e XP direto das Trincheiras. InfoQ, 2008. Disponível em: http://www.infoq.com/br/minibooks/scrum-xp-from-the-trenches . Acesso em: 9 de agosto de 2016.

Bibliografia Complementar
MARTIN, R. C. Agile Software Development, Principles, Patterns, and Practices. Prentice Hall, 2002
BOEG, J. Kanban em 10 Passos. InfoQ, 2012. Disponível em: http://www.infoq.com/br/minibooks/priming-kanbanjesper-boeg . Acesso em: 31 de outubro de 2013.
GOMES, André Faria. Agile: Desenvolvimento de software com entregas frequentes e foco no valor de negócio. Casa do Código, 2013.
BECK, Kent; ANDRES, Cynthia. Extreme programming explained: embrace change . 2nd ed. Boston, Mass.: Addison Wesley, 2005. xxii, 189 p. ISBN 9780321278654.
ANDERSON, D. J. Kanban. Blue Hole Press, 2010.
FOWLER, Martin. Refatoração: aperfeiçoando o projeto de código existente . Porto Alegre: Bookman, 2004. 365 p. ISBN 8536303956.
TELES, V. M. Extreme programming: aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade. São Paulo: Novatec Editora, 2006.
FREEMAN, S. PRYCE, N. Growing Object-Oriented Software, Guided by Tests. Addison-Wesley, 2009.

#	Resumo da Alteração	Edição	Data	Aprovação	Data
1	Tudo novo.	Adolfo Gustavo Serra Seca Neto	14/02/2017	Leonelo Dell Anhol Almeida	14/02/2017

