

Modulbezeichnung:	No-Frills Software Engineering	
Modulnummer: IWNF	Semester: --	Dauer: Minimaldauer 1 Semester
Modultyp: Wahlpflicht		Regulär angeboten im: WS, SS
Workload: 300 h		ECTS Punkte: 10
Zugangsvoraussetzungen: IT-Projektmanagement		Unterrichtssprache: Deutsch
Kurse im Modul: <ul style="list-style-type: none"> • Techniken und Methoden der agilen Softwareentwicklung (IWNF01) • Fallstudie Agiles Software Engineering (IWNF02) 		Workload: Selbststudium: 210 h Selbstüberprüfung: 30 h Tutorien: 60 h
Kurskoordinatoren/Tutoren:: Siehe aktuelle Liste der Tutoren im Learning Management System		Modulverantwortliche(r): Prof. Dr. Ralf Kneuper
Bezüge zu anderen Programmen: keine		Bezüge zu anderen Modulen im Programm: <ul style="list-style-type: none"> • Requirements Engineering • Spezifikation • IT-Projektmanagement • Qualitätssicherung im Softwareprozess
Qualifikations- und Lernziele des Moduls:		
Techniken und Methoden der agilen Softwareentwicklung: Nach erfolgreichem Abschluss sind die Studierenden in der Lage, <ul style="list-style-type: none"> • Probleme und Risiken der industriellen SW-Entwicklung und ihre Konsequenzen für Entwicklungsprozesse zu analysieren und beurteilen. • die Grundprinzipien des „No-Frills Software Engineering“ zu kennen und zu verstehen. • Praxisszenarien zu analysieren und selbständig geeignete Methoden und Werkzeuge des „No-Frills Software Engineering“ anzuwenden. 		
Projekt Agiles Software-Engineering: Nach erfolgreichem Abschluss sind die Studierenden in der Lage, <ul style="list-style-type: none"> • typische Probleme in verschiedenen Projektsituationen durch den gezielten Einsatz von agilen Techniken und Methoden zu adressieren. • die Konzeption und den projektspezifischen Einsatz von Techniken und Werkzeugen zu dokumentieren. 		
Lehrinhalt des Moduls:		

Techniken und Methoden der agilen Softwareentwicklung:

- Merkmale und Prinzipien von Agilität
- Agilität in kleinen Teams mit Scrum
- Agiles Portfolio- und Projektmanagement
- Agiles Anforderungs- und IT-Architekturmanagement
- Agiles Testen
- Agile Delivery and Deployment

Projekt Agiles Software-Engineering:

Umsetzung und Dokumentation von Projekten zum Einsatz agiler Techniken sowie Vertiefung ausgewählter Themengebiete aus dem Kurs IWNF01.

Lehrmethoden:	Siehe Kursbeschreibungen	
Literatur:	Siehe Literaturlisten der vorliegenden Kursbeschreibungen	
Anteil der Modulnote an der Gesamtabschlussnote des Programms: --	Prüfungszulassungsvoraussetzung:	Abschlussprüfungen:
	Siehe Kursbeschreibungen	IWNF01: Klausur, 90 Min. (50%) IWNF02: Schriftliche Ausarbeitung: Projektbericht (50%)

Kursnummer: IWNF01	Kursname: Techniken und Methoden der agilen Softwareentwicklung	Gesamtstunden: 150 h
		ECTS Punkte: 5 ECTS
Kurstyp: Wahlpflicht Kursangebot: Kursdauer: Minimaldauer 1 Semester	Zugangsvoraussetzungen: Spezifikation, IT-Projektmanagement	
Kurskoordinator(en) / Dozenten / Lektoren: Siehe aktuelle Liste der Tutoren im Learning Management System	Bezüge zu anderen Modulen: Siehe Modulbeschreibung	
<p>Beschreibung des Kurses:</p> <p>Ziel des Kurses ist es, den Studierenden einen vertiefenden Einblick in das Thema agile Softwareentwicklung zu vermitteln. Dazu werden zunächst die grundlegenden Merkmale und Prinzipien von Agilität vorgestellt und diskutiert. Danach wird dargestellt, wie kleine Projekte und Teams agiles Software Engineering betreiben können und wie sich die agilen Prinzipien auf große Projekte übertragen und dort anwenden lassen. Anschließend werden agile Techniken für ausgewählte Kernaktivitäten im Software Engineering vermittelt, wobei ein Schwerpunkt auf dem Gebiet Testen, Delivery und Deployment liegt.</p> <p>Kursziele:</p> <p>Nach erfolgreichem Abschluss des Kurses</p> <ul style="list-style-type: none"> • können die Studierenden Probleme und Risiken der industriellen SW-Entwicklung und ihre Konsequenzen für Entwicklungsprozesse analysieren und beurteilen. • kennen und verstehen die Studierenden die Grundprinzipien des „No-Frills Software Engineering“. • können die Studierenden Praxisszenarien analysieren und selbständig geeignete Methoden und Werkzeuge des „No-Frills Software Engineering“ anwenden. <p>Lehrmethoden:</p> <p>Die Lehrmaterialien enthalten einen kursabhängigen Mix aus Skripten, Video-Vorlesungen, Übungen, Podcasts, (Online-)Tutorien, Fallstudien. Sie sind so strukturiert, dass Studierende sie in freier Ortswahl und zeitlich unabhängig bearbeiten können.</p> <p>Inhalte des Kurses:</p> <p>1 Merkmale und Prinzipien von Agilität</p> <p>1.1 Merkmale und Herausforderungen von Softwareprojekten 1.2 Klassifikationen von Unsicherheit 1.3 Gegenüberstellung von agiler und klassischer Softwareentwicklung 1.4 Prinzipien der Agilität</p> <p>2 Agilität in kleinen Teams mit Scrum</p> <p>2.1 Grundlagen und allgemeiner Aufbau von Scrum 2.2 Zentrales Managementartefakt: Product Backlog 2.3 Weitere Managementartefakte</p> <p>3 Agiles Portfolio- und Projektmanagement</p> <p>3.1 Planungsebenen im agilen Projektmanagement 3.2 Agiles Portfoliomanagement</p>		

3.3 Organisation mehrerer Teams in einem Projekt

3.4 Produkt- und Releaseplanung

4 Agiles Anforderungs- und IT-Architekturmanagement

4.1 Requirements Engineering in agilen Projekten

4.2 Architekturmanagement in agilen Projekten

5 Agiles Testen

5.1 Grundlagen und Anforderungen an die QS-Organisation

5.2 Teststufen und Agilität

5.3 Testautomatisierung

6 Agile Delivery and Deployment

6.1 Grundlagen und Continuous Delivery Pipeline

6.2 Continuous Build und Continuous Integration

6.3 Akzeptanztests, Lasttests und Continuous Deployment

Literatur:

- Baumgartner, M. et al. (2013): Agile Testing. Der agile Weg zur Qualität. Hanser, München. ISBN-13: 978-3446431942.
- Biffi, S. et al. (Hrsg.) (2005): Value-Based Software Engineering. Springer, Berlin/Heidelberg. ISBN-13: 978-3540259930.
- Cockburn, A. (2007): Agile Software Development. The Cooperative Game, 2. Auflage, Addison-Wesley, Upper Saddle River (NJ). ISBN-13: 978-0321482754.
- DeMarco, T. (2003): Bärenango. Mit Risikomanagement Projekte zum Erfolg führen. Hanser, München. ISBN-13: 978-3446223332.
- Epping, T. (2011): Kanban für die Softwareentwicklung. Springer, Berlin/Heidelberg. ISBN-13: 978-3642225949.
- Frehr, H.-U. (1994): Total Quality Management. Unternehmensweite Qualitätsverbesserung. Hanser, München. ISBN-13: 978-3446177796.
- Geirhos, M. (2011): IT-Projektmanagement. Was wirklich funktioniert – und was nicht. Galileo Computing, Bonn. ISBN-13: 978-3836217736.
- Höhn, R./Höppner, S. (2008): Das V-Modell XT. Springer, Heidelberg/Berlin. ISBN-13: 978-3540302490.
- Hummel, H. (2011): Aufwandsschätzungen in der Software- und Systementwicklung. Spektrum, Wiesbaden. ISBN-13: 978-3827427519.
- Künne, T. (2012): Android 4: Apps entwickeln mit dem Android SDK. Galileo Computing, Bonn. ISBN-13: 978-3836219488.
- Link, J. (2005): Softwaretests mit JUnit. 2.Auflage, dpunkt.verlag, Heidelberg. ISBN-13: 978-3898643252.
- Mangold, P. (2009): IT-Projektmanagement. 3. Auflage, Spektrum, Wiesbaden. ISBN-13: 978-3827419378.
- McConnell, S. (2006): Software Estimation. Demystifying the Black Art. Microsoft Press, Redmond (WA). ISBN-13: 978-0735605350.
- Motzel, E./O. Pannenbäcker (1998): Projektmanagement-Kanon. Der deutsche Zugang zum Project Management Body of Knowledge. TÜV-Verlag, Köln. ISBN-13: 978-3824904983.
- Pichler, R. (2007): Scrum. Agiles Projektmanagement erfolgreich einsetzen; dpunkt.verlag, Heidelberg. (2007) ISBN 978-3898644785
- Röpstorff, S./Wiechmann, R. (2012): Scrum in der Praxis. Erfahrungen, Problemfelder und Erfolgsfaktoren. dpunkt.verlag, Heidelberg. ISBN-13: 978-3898647922.
- Rubin, K. S. (2014): Essential Scrum. Umfassendes Scrum-Wissen aus der Praxis. mitp, Frechen. ISBN-13: 978-3826690471.
- Schwalbe, K. (2010): Information Technology Project Management. 6. Auflage, Cengage, Boston. ISBN-13: 978-1111221751.
- Tiemeyer, E. (2010): Handbuch IT-Projektmanagement, Vorgehensmodelle, Managementinstrumente, Good Practices. Hanser, München. ISBN-13: 978-3446421929.
- Versteegen, G. (2000): Projektmanagement. Mit dem Rational Unified Process. Springer, Heidelberg/Berlin. ISBN-13: 978-3540667551.
- Wirdemann, R. (2011): Scrum mit User Stories. 2. Auflage, Hanser, München. ISBN-13: 978-3446426603.
- Wolff, E. (2014): Continuous Delivery. Der pragmatische Einstieg. dpunkt.verlag, Heidelberg. ISBN-13: 978-3864902086.
- Wolf, H./Bleek/W.-G. (2010): Agile Softwareentwicklung. Werte, Konzepte und Methoden. 2. Auflage, dpunkt.verlag, Heidelberg. ISBN-13: 978-3898647014.

Prüfungsleistung:

Klausur, 90 Min.

Zeitaufwand Studierenden (in Std.): 150

Selbststudium (in Std.): 90

Selbstüberprüfung (in Std.): 30

Tutorien (in Std.): 30

Kursnummer: IWNF02	Kursname: Fallstudie Agiles Software Engineering	Gesamtstunden: 150 h
		ECTS Punkte: 5 ECTS
Kurstyp: Wahlpflicht Kursangebot: Kursdauer: Minimaldauer 1 Semester	Zugangsvoraussetzungen: Spezifikation, IT-Projektmanagement	
Kurskoordinator(en) / Dozenten / Lektoren: Siehe aktuelle Liste der Tutoren im Learning Management System	Bezüge zu anderen Modulen: Siehe Modulbeschreibung	
<p>Beschreibung des Kurses:</p> <p>Aufbauend auf den im Kurs „Techniken und Methoden der agilen SW-Entwicklung“ vermittelten Inhalten lösen die Studierenden in diesem Kurs selbstständig praktische Fallstudien in diesem Themengebiet und dokumentieren ihr Ergebnis in Form einer schriftlichen Hausarbeit.</p> <p>Kursziele:</p> <p>Nach erfolgreichem Abschluss des Kurses</p> <ul style="list-style-type: none"> • können die Studierenden typische Probleme in verschiedenen Projektsituationen durch den gezielten Einsatz von agilen Techniken und Methoden adressieren. • können die Studierenden die Konzeption und den projektspezifischen Einsatz von Techniken und Werkzeugen dokumentieren. <p>Lehrmethoden:</p> <p>Die Lehrmaterialien enthalten einen kursabhängigen Mix aus Skripten, Video-Vorlesungen, Übungen, Podcasts, (Online-)Tutorien, Fallstudien. Sie sind so strukturiert, dass Studierende sie in freier Ortswahl und zeitlich unabhängig bearbeiten können.</p> <p>Inhalte des Kurses:</p> <p>Umsetzung und Dokumentation von praktischen Fallstudien und Szenarien zum Einsatz agiler Techniken sowie Vertiefung ausgewählter Themengebiete aus dem Kurs IWNF01.</p>		

Literatur:

- Allweyer, T. (2009): BPMN 2.0. Business Process Model and Notation. Einführung in den Standard für die Geschäftsprozessmodellierung. 2. Auflage, Books on Demand, Norderstedt. ISBN-13: 978-3839121344.
- Balzert, H. (2004): Lehrbuch der Objektmodellierung. Analyse und Entwurf mit der UML 2. 2. Auflage, Spektrum, Heidelberg/Berlin. ISBN-13: 978-3827411624.
- Balzert, H. (2010): UML 2 kompakt mit Checklisten. 3. Auflage, Spektrum, Heidelberg. ISBN-13: 978-3827425065.
- Baumgartner, M. et al. (2013): Agile Testing. Der agile Weg zur Qualität. Hanser, München. ISBN-13: 978-3446431942.
- Booch, G./Rumbaugh, J./Jacobson, I. (2006): Das UML Benutzerhandbuch. Addison-Wesley, Bonn. ISBN-13: 978-3827322951.
- Cohn, M. (2010): User Stories für die agile Software-Entwicklung mit Scrum, XP u.a. mitp, Frechen. ISBN-13: 978-3826658983.
- DeMarco, T. (2003): Bärenango. Mit Risikomanagement Projekte zum Erfolg führen. Hanser, München. ISBN-13: 978-3446223332.
- Epping, T. (2011): Kanban für die Softwareentwicklung. Springer, Berlin/Heidelberg. ISBN-13: 978-3642225949.
- Freund, J./Rücker, B. (2012): Praxishandbuch BPMN 2.0. 3. Auflage, Hanser. München. ISBN-13: 978-3446429864.
- Gadatsch, A. (2012): Grundkurs Geschäftsprozess-Management. Methoden und Werkzeuge für die IT-Praxis: Eine Einführung für Studenten und Praktiker. 7. Auflage, Vieweg+Teubner, Wiesbaden. ISBN-13: 978-3834824271.
- Mangold, P. (2009): IT-Projektmanagement. 3. Auflage, Spektrum, Wiesbaden. ISBN-13: 978-3827419378.
- Röpstorff, S./Wiechmann, R. (2012): Scrum in der Praxis. Erfahrungen, Problemfelder und Erfolgsfaktoren. dpunkt.verlag, Heidelberg. ISBN-13: 978-3898647922.
- Rubin, K. S. (2014): Essential Scrum. Umfassendes Scrum-Wissen aus der Praxis. mitp, Frechen. ISBN-13: 978-3826690471.
- Wirdemann, R. (2011): Scrum mit User Stories. 2. Auflage, Hanser, München. ISBN-13: 978-3446426603.
- Wolff, E. (2014): Continuous Delivery. Der pragmatische Einstieg. dpunkt.verlag, Heidelberg. ISBN-13: 978-3864902086.
- Wolf, H./Bleek/W.-G. (2010): Agile Softwareentwicklung. Werte, Konzepte und Methoden. 2. Auflage, dpunkt.verlag, Heidelberg. ISBN-13: 978-3898647014.

Prüfungsleistung:

Schriftliche Ausarbeitung:
Fallstudie

Zeitaufwand Studierenden (in Std.): 150

Selbststudium (in Std.): 110
Selbstüberprüfung (in Std.): 20
Tutorien (in Std.): 20