

MODULHANDBUCH

Agile Java Web Developer (IU)

Weiterbildung Agile Java Web Developer (UPS-AJWD)

n/a ECTS

Fernstudium

Klassifizierung: Diploma

Inhaltsverzeichnis

1. Semester

Modul IPWA1: Programmierung von Web-Anwendungsoberflächen

Modulbeschreibung7

Kurs IPWA01: Programmierung von Web-Anwendungsoberflächen9

Modul IPWA2: Programmierung von industriellen Informationssystemen mit Java EE

Modulbeschreibung15

Kurs IPWA02: Programmierung von industriellen Informationssystemen mit Java EE17

Modul IWNF1: Techniken und Methoden der agilen Softwareentwicklung

Modulbeschreibung23

Kurs IWNF01: Techniken und Methoden der agilen Softwareentwicklung25

Modul IWNF2: Projekt Agiles Software Engineering

Modulbeschreibung29

Kurs IWNF02: Projekt Agiles Software Engineering31

1. Semester

Programmierung von Web-Anwendungsoberflächen

Modulcode: IPWA1

Modultyp	Zugangsvoraussetzungen	Niveau	ECTS	Zeitaufwand Studierende
s. Curriculum	keine	BA	n/a	150 h

Semester	Dauer	Regulär angeboten im	Unterrichtssprache
s. Curriculum	Minimaldauer: 1 Semester	WiSe/SoSe	Deutsch

Modulverantwortliche(r)

Prof. Dr. Marian Benner-Wickner (Programmierung von Web-Anwendungsoberflächen)

Kurse im Modul

- Programmierung von Web-Anwendungsoberflächen (IPWA01)

Art der Prüfung(en)

Modulprüfung

Studienformat: Kombistudium
Klausur, 90 Minuten

Studienformat: Fernstudium
Klausur, 90 Minuten

Teilmodulprüfung

Anteil der Modulnote an der Gesamtnote

s. Curriculum

Lehrinhalt des Moduls

- Aufbau und Überblick von Web-Architekturen
- Statische Webseiten
- Grundlegende Java-Web-Technologien
- Komponentenbasierte Web-Benutzerschnittstellen
- Verknüpfung von View und Model
- Komponentenbibliotheken

Qualifikationsziele des Moduls**Programmierung von Web-Anwendungsoberflächen**

Nach erfolgreichem Abschluss sind die Studierenden in der Lage,

- die Architektur von Enterprise-Web-Anwendungen wiederzugeben sowie deren Aufbau und deren Zusammenspiel zu erläutern.
- relevante Technologien, Standards und Frameworks zur Erstellung von Anwendungsoberflächen von Enterprise-Web-Anwendungen zu beschreiben und sie voneinander abzugrenzen.
- Web-Technologien zur Implementierung von Anwendungsoberflächen zu verwenden.
- selbstständig geeignete Oberflächen zur Lösung von konkret beschriebenen Problemen von Web-Anwendungen zu erstellen.

Bezüge zu anderen Modulen im Studiengang

Ist Grundlage für weitere Module im Bereich Informatik & Software-Entwicklung

Bezüge zu anderen Studiengängen der IUBH

Alle Bachelor-Programme im Bereich IT & Technik

Programmierung von Web-Anwendungsoberflächen

Kurscode: IPWA01

Niveau	Unterrichtssprache	SWS	ECTS	Zugangsvoraussetzungen
BA	Deutsch		n/a	keine

Beschreibung des Kurses

Auf Basis der im Modul „Objektorientierte Programmierung“ vermittelten Kenntnisse vermittelt dieses Modul Kenntnisse und Erfahrungen im Aufbau und der Erstellung von webbasierten betrieblichen Informationssystemen. Dabei lernen die Studierenden die verschiedenen Architekturschichten eines Informationssystems (Oberfläche, Geschäftslogik, Datenschicht) anhand eines typischen Java-Technologie-Stacks kennen. Neben dem allgemeinen Aufbau von Web-Anwendungen steht in diesem Kurs die Erstellung von Web-Oberflächen im Vordergrund: Die Studierenden lernen, wie die Web-Anwendungsoberflächen von IT-Systemen gebaut werden und welche Technologien dabei typischerweise zum Einsatz kommen.

Kursziele

Nach erfolgreichem Abschluss sind die Studierenden in der Lage,

- die Architektur von Enterprise-Web-Anwendungen wiederzugeben sowie deren Aufbau und deren Zusammenspiel zu erläutern.
- relevante Technologien, Standards und Frameworks zur Erstellung von Anwendungsoberflächen von Enterprise-Web-Anwendungen zu beschreiben und sie voneinander abzugrenzen.
- Web-Technologien zur Implementierung von Anwendungsoberflächen zu verwenden.
- selbstständig geeignete Oberflächen zur Lösung von konkret beschriebenen Problemen von Web-Anwendungen zu erstellen.

Kursinhalt

1. Aufbau und Überblick von Web-Architekturen
 - 1.1 Client-Server, 3-Schichten-Architektur, Model-View-Controller
 - 1.2 Typische Java Enterprise (Java EE)-Architekturen
2. Statische Webseiten
 - 2.1 XHTML
 - 2.2 CSS
 - 2.3 XML

3. Grundlegende Java-Web-Technologien
 - 3.1 Java-Servlets
 - 3.2 Java Server Pages (JSP)
 - 3.3 AJAX
4. Komponentenbasierte Web-Benutzerschnittstellen
 - 4.1 Einführung in Java Server Faces (JSF)
 - 4.2 JSF-Komponenten
5. Verknüpfung von View und Model
 - 5.1 Managed Beans
 - 5.2 Unified Expression Language (UEL)
6. Komponentenbibliotheken
 - 6.1 Komponentenbibliotheken
 - 6.2 PrimeFaces
 - 6.3 A4

Literatur**Pflichtliteratur****Weiterführende Literatur**

- Balzert, H. (2011): Basiswissen Web-Programmierung. XHTML, CSS, JavaScript, XML, PHP, JSP, ASP.NET, Ajax. 2. Auflage, W3L, Dortmund.
- Ihns, O. et al. (2011): EJB 3.1 professionell. Grundlagen- und Expertenwissen zu Enterprise JavaBeans 3.1. 2. Auflage, dpunkt.verlag, Heidelberg.
- Koch, S. (2011): JavaScript. Einführung, Programmierung und Referenz. 6. Auflage, dpunkt.verlag, Heidelberg.
- Leßner, J./Eberling, W. (2011): Enterprise JavaBeans 3.1. Das EJB-Praxisbuch für Ein- und Umsteiger. 2. Auflage, Hanser, München.
- Marinschek, M./Kurz, M./Müllan, G. (2009): JavaServer Faces 2.0. Grundlagen und erweiterte Konzepte. 2. Auflage, dpunkt.verlag, Heidelberg.
- Müller, B. (2010): JavaServer Faces 2.0. Ein Arbeitsbuch für die Praxis. 2. Auflage, Hanser, München.
- Originale Dokumentationen und Material zu Java EE: (URL: <http://www.oracle.com/technetwork/java/javaee/documentation/apis-139520.html> [letzter Zugriff: 27.02.2017]).
- Originale Dokumentationen und Material zu Java-Server Faces (JSF): (URL: <http://www.oracle.com/technetwork/java/javaee/documentation/index-137726.html> [letzter Zugriff: 27.02.2017]).
- Popp, G. (2009): Konfigurationsmanagement mit Subversion, Maven und Redmine. Grundlagen für Softwarearchitekten und Entwickler. 5. Auflage, dpunkt.verlag, Heidelberg.
- Vonhoegen, H. (2011): Einstieg in XML. Grundlagen, Praxis, Referenzen. 6. Auflage, Galileo Computing, Bonn.
- Weil, D. (2012): Java EE 6. Enterprise-Anwendungsentwicklung leicht gemacht. entwickler.press, Frankfurt a. M.

Studienformat Kombistudium

Studienform Kombistudium	Kursart Vorlesung
------------------------------------	-----------------------------

Informationen zur Prüfung	
Prüfungszulassungsvoraussetzungen	BOLK: Ja Evaluation: Nein
Prüfungsleistung	Klausur, 90 Minuten

Zeitaufwand Studierende					
Selbststudium	Präsenzstudium	Tutorium	Selbstüberprüfung	Praxisanteil	Gesamt
90 h	0 h	30 h	30 h	0 h	150 h

Lehrmethoden	
<input type="checkbox"/> Learning Sprints® <input checked="" type="checkbox"/> Skript <input type="checkbox"/> Vodcast <input checked="" type="checkbox"/> Shortcast <input checked="" type="checkbox"/> Audio <input checked="" type="checkbox"/> Musterklausur	<input type="checkbox"/> Repetitorium <input type="checkbox"/> Creative Lab <input type="checkbox"/> Prüfungsleitfaden <input checked="" type="checkbox"/> Live Tutorium/Course Feed

Studienformat Fernstudium

Studienform Fernstudium	Kursart Online-Vorlesung
-----------------------------------	------------------------------------

Informationen zur Prüfung	
Prüfungszulassungsvoraussetzungen	BOLK: Ja Evaluation: Nein
Prüfungsleistung	Klausur, 90 Minuten

Zeitaufwand Studierende					
Selbststudium	Präsenzstudium	Tutorium	Selbstüberprüfung	Praxisanteil	Gesamt
90 h	0 h	30 h	30 h	0 h	150 h

Lehrmethoden	
<input type="checkbox"/> Learning Sprints® <input checked="" type="checkbox"/> Skript <input type="checkbox"/> Vodcast <input checked="" type="checkbox"/> Shortcast <input checked="" type="checkbox"/> Audio <input checked="" type="checkbox"/> Musterklausur	<input type="checkbox"/> Repetitorium <input type="checkbox"/> Creative Lab <input type="checkbox"/> Prüfungsleitfaden <input checked="" type="checkbox"/> Live Tutorium/Course Feed

IPWA01

Programmierung von industriellen Informationssystemen mit Java EE

Modulcode: IPWA2

Modultyp	Zugangsvoraussetzungen	Niveau	ECTS	Zeitaufwand Studierende
s. Curriculum	keine	BA	n/a	150 h

Semester	Dauer	Regulär angeboten im	Unterrichtssprache
s. Curriculum	Minimaldauer: 1 Semester	WiSe/SoSe	Deutsch

Modulverantwortliche(r)

Prof. Dr. Marian Benner-Wickner (Programmierung von industriellen Informationssystemen mit Java EE)

Kurse im Modul

- Programmierung von industriellen Informationssystemen mit Java EE (IPWA02)

Art der Prüfung(en)

Modulprüfung

Studienformat: Kombistudium

Klausur, 90 Minuten

Studienformat: Fernstudium

Klausur, 90 Minuten

Teilmodulprüfung

Anteil der Modulnote an der Gesamtnote

s. Curriculum

Lehrinhalt des Moduls

- Programmierung von Geschäftslogik
- Programmierung von Datenbankanbindungen

Qualifikationsziele des Moduls**Programmierung von industriellen Informationssystemen mit Java EE**

Nach erfolgreichem Abschluss sind die Studierenden in der Lage,

- relevante Technologien, Standards und Frameworks zur Erstellung von Geschäftslogik und Datenbankanbindung in Enterprise-Web-Anwendungen zu erläutern und voneinander abzugrenzen.
- verschiedenen Web-Technologien zur Implementierung und Integration von Anwendungsoberflächen, Geschäftslogik und Datenschicht zu verwenden.
- selbstständig geeignete Web-Anwendungen zur Lösung von konkret beschriebenen Problemen zu erstellen.

Bezüge zu anderen Modulen im Studiengang

Ist Grundlage für weitere Module im Bereich Informatik & Software-Entwicklung

Bezüge zu anderen Studiengängen der IUBH

Alle Bachelor-Programme im Bereich IT & Technik

Programmierung von industriellen Informationssystemen mit Java EE

Kurscode: IPWA02

Niveau	Unterrichtssprache	SWS	ECTS	Zugangsvoraussetzungen
BA	Deutsch		n/a	keine

Beschreibung des Kurses

In diesem Kurs die Geschäftslogik und die Datenbankanbindung von Web-Anwendungen im Fokus. Zunächst werden einfache Web-Oberflächen um Funktionen zu Validierung und Konvertierung von Eingaben erweitert und Fehlermeldungen erstellt. Anschließend werden Konzepte zur Umsetzung von Navigationsstrukturen vermittelt. Als letzte Schicht von Web-Anwendungen werden Konzepte und Technologien zur Anbindung an Datenbanksystemen vorgestellt und angewandt.

Kursziele

Nach erfolgreichem Abschluss sind die Studierenden in der Lage,

- relevante Technologien, Standards und Frameworks zur Erstellung von Geschäftslogik und Datenbankanbindung in Enterprise-Web-Anwendungen zu erläutern und voneinander abzugrenzen.
- verschiedenen Web-Technologien zur Implementierung und Integration von Anwendungsoberflächen, Geschäftslogik und Datenschicht zu verwenden.
- selbstständig geeignete Web-Anwendungen zur Lösung von konkret beschriebenen Problemen zu erstellen.

Kursinhalt

1. Programmierung von Geschäftslogik
 - 1.1 Validierung von Benutzereingaben
 - 1.2 Konvertierung von Zahlen, Währungen und Datumsfeldern
 - 1.3 Definition von Fehlermeldungen
 - 1.4 Seitennavigation mit JSF-Anwendungen
 - 1.5 JSF-Lebenszyklus und Eventhandling

2. Programmierung von Datenbankanbindungen
 - 2.1 Persistente Speicherung in Datenbanken
 - 2.2 Objektrelationales Mapping
 - 2.3 Datenbankpersistenz mit JPA und Hibernate als Persistence Provider
 - 2.4 Erzeugen von Entitäten
 - 2.5 Assoziationen zwischen Objekten mittels JPA
 - 2.6 Löschen von Objekten mittels JPA
 - 2.7 Data Access Objects

Literatur**Pflichtliteratur****Weiterführende Literatur**

- Balzert, H. (2011): Basiswissen Web-Programmierung. XHTML, CSS, JavaScript, XML, PHP, JSP, ASP.NET, Ajax. 2. Auflage, W3L, Dortmund.
- Ihns, O. et al. (2011): EJB 3.1 professionell. Grundlagen- und Expertenwissen zu Enterprise JavaBeans 3.1. 2. Auflage, dpunkt.verlag, Heidelberg.
- Kehle, M./Hien, R./Röder, D. (2010): Hibernate und die Java Persistence API. Einstieg und professioneller Einsatz. 2. Auflage, entwickler.press, Frankfurt a. M.
- Koch, S. (2011): JavaScript. Einführung, Programmierung und Referenz. 6. Auflage, dpunkt.verlag, Heidelberg.
- Leßner, J./Eberling, W. (2011): Enterprise JavaBeans 3.1. Das EJB-Praxisbuch für Ein- und Umsteiger. 2. Auflage, Hanser, München.
- Marinschek, M./Kurz, M./Müllan, G. (2009): JavaServer Faces 2.0. Grundlagen und erweiterte Konzepte. 2. Auflage, dpunkt.verlag, Heidelberg.
- Müller, B. (2010): JavaServer Faces 2.0. Ein Arbeitsbuch für die Praxis. 2. Auflage, Hanser, München.
- Müller, B./Wehr, H. (2012): Java Persistence API 2. Hibernate, EclipseLink, OpenJPA und Erweiterungen. Hanser, München.
- Originale Dokumentationen und Material zu Hibernate: (URL: www.hibernate.org [letzter Zugriff: 27.02.2017]).
- Originale Dokumentationen und Material zu Java EE: (URL: <http://www.oracle.com/technetwork/java/javaee/documentation/apis-139520.html> [letzter Zugriff: 27.02.2017]).
- Originale Dokumentationen und Material zu Java-Server Faces (JSF): (URL: <http://www.oracle.com/technetwork/java/javaee/documentation/index-137726.html> [letzter Zugriff: 27.02.2017]).
- Popp, G. (2009): Konfigurationsmanagement mit Subversion, Maven und Redmine. Grundlagen für Softwarearchitekten und Entwickler. 5. Auflage, dpunkt.verlag, Heidelberg.
- Vonhoegen, H. (2011): Einstieg in XML. Grundlagen, Praxis, Referenzen. 6. Auflage, Galileo Computing, Bonn.
- Weil, D. (2012): Java EE 6. Enterprise-Anwendungsentwicklung leicht gemacht. entwickler.press, Frankfurt a. M.

Studienformat Kombistudium

Studienform Kombistudium	Kursart Vorlesung
------------------------------------	-----------------------------

Informationen zur Prüfung	
Prüfungszulassungsvoraussetzungen	BOLK: Ja Evaluation: Nein
Prüfungsleistung	Klausur, 90 Minuten

Zeitaufwand Studierende					
Selbststudium	Präsenzstudium	Tutorium	Selbstüberprüfung	Praxisanteil	Gesamt
90 h	0 h	30 h	30 h	0 h	150 h

Lehrmethoden	
<input type="checkbox"/> Learning Sprints® <input checked="" type="checkbox"/> Skript <input type="checkbox"/> Vodcast <input checked="" type="checkbox"/> Shortcast <input checked="" type="checkbox"/> Audio <input checked="" type="checkbox"/> Musterklausur	<input type="checkbox"/> Repetitorium <input type="checkbox"/> Creative Lab <input type="checkbox"/> Prüfungsleitfaden <input checked="" type="checkbox"/> Live Tutorium/Course Feed

Studienformat Fernstudium

Studienform Fernstudium	Kursart Online-Vorlesung
-----------------------------------	------------------------------------

Informationen zur Prüfung	
Prüfungszulassungsvoraussetzungen	BOLK: Ja Evaluation: Nein
Prüfungsleistung	Klausur, 90 Minuten

Zeitaufwand Studierende					
Selbststudium 90 h	Präsenzstudium 0 h	Tutorium 30 h	Selbstüberprüfung 30 h	Praxisanteil 0 h	Gesamt 150 h

Lehrmethoden	
<input type="checkbox"/> Learning Sprints® <input checked="" type="checkbox"/> Skript <input type="checkbox"/> Vodcast <input checked="" type="checkbox"/> Shortcast <input checked="" type="checkbox"/> Audio <input checked="" type="checkbox"/> Musterklausur	<input type="checkbox"/> Repetitorium <input type="checkbox"/> Creative Lab <input type="checkbox"/> Prüfungsleitfaden <input checked="" type="checkbox"/> Live Tutorium/Course Feed

IPWA02

Techniken und Methoden der agilen Softwareentwicklung

Modulcode: IWNF1

Modultyp	Zugangsvoraussetzungen	Niveau	ECTS	Zeitaufwand Studierende
s. Curriculum	keine	BA	n/a	150 h

Semester	Dauer	Regulär angeboten im	Unterrichtssprache
s. Curriculum	Minimaldauer: 1 Semester	WiSe/SoSe	Deutsch

Modulverantwortliche(r)

Prof. Dr. Damir Ismailovic (Techniken und Methoden der agilen Softwareentwicklung)

Kurse im Modul

- Techniken und Methoden der agilen Softwareentwicklung (IWNF01)

Art der Prüfung(en)

Modulprüfung

Studienformat: Fernstudium
Klausur, 90 Minuten

Teilmodulprüfung

Anteil der Modulnote an der Gesamtnote

s. Curriculum

Lehrinhalt des Moduls

- 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

Qualifikationsziele des Moduls**Techniken und Methoden der agilen Softwareentwicklung**

Nach erfolgreichem Abschluss sind die Studierenden in der Lage,

- Probleme und Risiken der industriellen SW-Entwicklung und ihre Konsequenzen für Entwicklungsprozesse zu analysieren und zu beurteilen.
- die Grundprinzipien des „No-Frills Software Engineering“ zu erläutern.
- Praxisszenarien zu analysieren und selbständig geeignete Methoden und Werkzeuge des „No-Frills Software Engineering“ anzuwenden.

Bezüge zu anderen Modulen im Studiengang

Ist Grundlage für alle weiteren Module aus dem Bereich Informatik & Software-Entwicklung

Bezüge zu anderen Studiengängen der IUBH

Alle Bachelor-Programme im Bereich IT & Technik

Techniken und Methoden der agilen Softwareentwicklung

Kurscode: IWNF01

Niveau	Unterrichtssprache	SWS	ECTS	Zugangsvoraussetzungen
BA	Deutsch		n/a	keine

Beschreibung des Kurses

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 Projekt 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.

Kursziele

Nach erfolgreichem Abschluss sind die Studierenden in der Lage,

- Probleme und Risiken der industriellen SW-Entwicklung und ihre Konsequenzen für Entwicklungsprozesse zu analysieren und zu beurteilen.
- die Grundprinzipien des „No-Frills Software Engineering“ zu erläutern.
- Praxisszenarien zu analysieren und selbständig geeignete Methoden und Werkzeuge des „No-Frills Software Engineering“ anzuwenden.

Kursinhalt

1. Merkmale und Prinzipien von Agilität
 - 1.1 Merkmale und Herausforderungen von Softwareprojekten
 - 1.2 Klassifikation von Unsicherheit
 - 1.3 Gegenüberstellung von agiler und klassischer Softwareentwicklung
 - 1.4 Prinzipien von Agilität
2. Agilität in kleinen Teams mit Scrum
 - 2.1 Grundlagen und allgemeiner Aufbau mit Scrum
 - 2.2 Zentrales Managementartefakt: Product Backlog
 - 2.3 Weitere Managementartefakte

3. Agiles Portfolio- und Projektmanagement
 - 3.1 Planungsebenen im agilen Projektmanagement
 - 3.2 Agiles Portfoliomanagement
 - 3.3 Organisation mehrerer Teams in einem Projekt
 - 3.4 Produkt- und Release-Planung
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 and Continuous Integration
 - 6.3 Akzeptanztests, Lasttests und Continuous Deployment

Literatur**Pflichtliteratur****Weiterführende Literatur**

- Baumgartner, M. et al. (2013): Agile Testing. Der agile Weg zur Qualität. Hanser, München.
- Biffi, S. et al. (Hrsg.) (2005): Value-Based Software Engineering. Springer, Berlin/Heidelberg.
- Cockburn, A. (2007): Agile Software Development. The Cooperative Game. 2. Auflage, Addison-Wesley, Upper Saddle River (NJ).
- DeMarco, T. (2003): Bärenango. Mit Risikomanagement Projekte zum Erfolg führen. Hanser, München.
- Epping, T. (2011): Kanban für die Softwareentwicklung. Springer, Berlin/Heidelberg.
- Geirhos, M. (2011): IT-Projektmanagement. Was wirklich funktioniert – und was nicht. Galileo Computing, Bonn.
- Hummel, H. (2011): Aufwandsschätzungen in der Software- und Systementwicklung. Spektrum, Wiesbaden.
- Künneth, T. (2012): Android 4. Apps entwickeln mit dem Android SDK. Galileo Computing, Bonn.
- Link, J. (2005): Softwaretests mit JUnit. 2.Auflage, dpunkt.verlag, Heidelberg.
- Mangold, P. (2009): IT-Projektmanagement. 3. Auflage, Spektrum, Wiesbaden.
- Motzel, E./O. Pannenbäcker (1998): Projektmanagement-Kanon. Der deutsche Zugang zum Project Management Body of Knowledge. TÜV-Verlag, Köln.
- Pichler, R. (2007): Scrum. Agiles Projektmanagement erfolgreich einsetzen. dpunkt.verlag, Heidelberg. (2007)
- Röpstorff, S./Wiechmann, R. (2012): Scrum in der Praxis. Erfahrungen, Problemfelder und Erfolgsfaktoren. dpunkt.verlag, Heidelberg.
- Rubin, K. S. (2014): Essential Scrum. Umfassendes Scrum-Wissen aus der Praxis. mitp, Frechen.
- Tiemeyer, E. (2010): Handbuch IT-Projektmanagement, Vorgehensmodelle, Managementinstrumente, Good Practices. Hanser, München.
- Wirdemann, R. (2011): Scrum mit User Stories. 2. Auflage, Hanser, München.
- Wolff, E. (2014): Continuous Delivery. Der pragmatische Einstieg. dpunkt.verlag, Heidelberg.
- Wolf, H./Bleek/W.-G. (2010): Agile Softwareentwicklung. Werte, Konzepte und Methoden. 2. Auflage, dpunkt.verlag, Heidelberg.

Studienformat Fernstudium

Studienform Fernstudium	Kursart Online-Vorlesung
-----------------------------------	------------------------------------

Informationen zur Prüfung	
Prüfungszulassungsvoraussetzungen	BOLK: Ja Evaluation: Nein
Prüfungsleistung	Klausur, 90 Minuten

Zeitaufwand Studierende					
Selbststudium	Präsenzstudium	Tutorium	Selbstüberprüfung	Praxisanteil	Gesamt
90 h	0 h	30 h	30 h	0 h	150 h

Lehrmethoden	
<input type="checkbox"/> Learning Sprints® <input checked="" type="checkbox"/> Skript <input type="checkbox"/> Vodcast <input checked="" type="checkbox"/> Shortcast <input checked="" type="checkbox"/> Audio <input checked="" type="checkbox"/> Musterklausur	<input type="checkbox"/> Repetitorium <input type="checkbox"/> Creative Lab <input type="checkbox"/> Prüfungsleitfaden <input type="checkbox"/> Live Tutorium/Course Feed

Projekt Agiles Software Engineering

Modulcode: IWNF2

Modultyp s. Curriculum	Zugangsvoraussetzungen keine	Niveau BA	ECTS n/a	Zeitaufwand Studierende 150 h
----------------------------------	----------------------------------------	---------------------	--------------------	-----------------------------------------

Semester s. Curriculum	Dauer Minimaldauer: 1 Semester	Regulär angeboten im WiSe/SoSe	Unterrichtssprache Deutsch
----------------------------------	---------------------------------------------	------------------------------------------	--------------------------------------

Modulverantwortliche(r)

Prof. Dr. Damir Ismailovic (Projekt Agiles Software Engineering)

Kurse im Modul

- Projekt Agiles Software Engineering (IWNF02)

Art der Prüfung(en)

Modulprüfung

Studienformat: Fernstudium
Schriftliche Ausarbeitung: Projektbericht

Teilmodulprüfung

Anteil der Modulnote an der Gesamtnote

s. Curriculum

Lehrinhalt des Moduls

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

Qualifikationsziele des Moduls**Projekt Agiles Software Engineering**

Nach erfolgreichem Abschluss sind die Studierenden in der Lage,

- 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.

Bezüge zu anderen Modulen im Studiengang

Baut auf Modulen aus dem Bereich Informatik & Software-Entwicklung auf

Bezüge zu anderen Studiengängen der IUBH

Alle Bachelor-Programme im Bereich IT & Technik

Projekt Agiles Software Engineering

Kurscode: IWNF02

Niveau	Unterrichtssprache	SWS	ECTS	Zugangsvoraussetzungen
BA	Deutsch		n/a	keine

Beschreibung des Kurses

Aufbauend auf ihren Kenntnissen im Bereich der agilen Softwareentwicklung setzen die Studierenden in diesem Kurs selbstständig Projekte in diesem Themengebiet um und dokumentieren ihr Ergebnis in Form einer schriftlichen Ausarbeitung.

Kursziele

Nach erfolgreichem Abschluss sind die Studierenden in der Lage,

- 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.

Kursinhalt

- Umsetzung und Dokumentation eines Projektes unter Einsatz agiler Techniken sowie Vertiefung der Kenntnisse im Bereich agiler Softwareentwicklung.

Literatur

Pflichtliteratur

Weiterführende Literatur

- Pichler, R. (2007): Scrum. Agiles Projektmanagement erfolgreich einsetzen. dpunkt.verlag, Heidelberg.
- Röpstorff, S./Wiechmann, R. (2012): Scrum in der Praxis. Erfahrungen, Problemfelder und Erfolgsfaktoren. dpunkt.verlag, Heidelberg.

Studienformat Fernstudium

Studienform Fernstudium	Kursart Projekt
-----------------------------------	---------------------------

Informationen zur Prüfung	
Prüfungszulassungsvoraussetzungen	BOLK: Nein Evaluation: Nein
Prüfungsleistung	Schriftliche Ausarbeitung: Projektbericht

Zeitaufwand Studierende					
Selbststudium 120 h	Präsenzstudium 0 h	Tutorium 30 h	Selbstüberprüfung 0 h	Praxisanteil 0 h	Gesamt 150 h

Lehrmethoden	
<input type="checkbox"/> Learning Sprints® <input type="checkbox"/> Skript <input type="checkbox"/> Vodcast <input type="checkbox"/> Shortcast <input type="checkbox"/> Audio <input type="checkbox"/> Musterklausur	<input type="checkbox"/> Repetitorium <input type="checkbox"/> Creative Lab <input checked="" type="checkbox"/> Prüfungsleitfaden <input type="checkbox"/> Live Tutorium/Course Feed