



Hochschule Reutlingen
Reutlingen University



Modulhandbuch

Digital Business Engineering

Fakultät Informatik
Hochschule Reutlingen

Wahlfächer im Studiengang

Master:

Digital Business Engineering (DBE)



Modulbeschreibung:

Elective: Innovative Methods of Service Engineering.....	3
Elective: Internet of Things (IoT).....	6
Elective: Scientific Publishing	8
Elective: Software Process Engineering	11
Elective: Agile Project Management and Software Development.....	14
Elective: Software Integration	16
Elective: Applied Machine Learning.....	19
Elective: Cloud-based Web Application Development	21
Elective: Algorithmics	23
Elective: Data Management & Analysis	25

Im Folgenden werden die einzelnen Wahlfächer im Detail beschrieben. Wird nichts anderes erwähnt, sind die zu erbringenden Prüfungsleistungen benotet.

Modul:	Elective: Innovative Methods of Service Engineering	
Kürzel:	DBE21, DBE31	
Lehrveranstaltung:	Innovative Methods of Service Engineering	
Veranstaltungsformat:	Vorlesung mit integrierter Übung	
Studiensemester:	Sommer- und Wintersemester	
Modulverantwortlicher:	Prof. Dr. Dieter Hertweck	
Dozent(in):	Prof. Dr. Dieter Hertweck	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Digital Business Engineering, Wahlfach, 2. und 3.Semester	
Lehrform / SWS:	Vorlesung & Übung	4 SWS
Arbeitsaufwand:	Präsenzstudium:	60 Stunden
	Eigenstudium:	120 Stunden
Kreditpunkte:	6 ECTS	
Voraussetzungen nach StuPro:		
Empfohlene Voraussetzungen:	Gute Betriebswirtschafts und Informatikkenntnisse; Kenntnisse im wissenschaftlichen Arbeiten	
Studien-/ Prüfungsleistungen/ Prüfungsform:	Vorlesung mit integrierten Übungen; Prüfungsleistungen siehe StuPro	

Modulziele

Im internationalen Wettbewerb werden zunehmend Produkte und Services honoriert, die sich zur Konkurrenz in Nutzwert oder Preis dramatisch differenzieren. Die Möglichkeiten eine solche Differenzierung in kürzester Zeit zu erreichen sind vielfältig wie nie. Neue Technologien und Paradigmen wie „in memory computing“, Cloud Computing, cyber-physische Systeme oder Soziale Netzwerke lassen binnen kürzester Zeit vollkommen neue Servicekreationen entstehen. Die Schöpfer solcher Services kommen aus unterschiedlichsten Disziplinen wie den Ingenieurwissenschaften, der Wirtschaftsinformatik oder den Sozialwissenschaften. Sie wenden bei der Service-Entwicklung z.T. komplett unterschiedliche Methoden an. Was z.Zt. fehlt ist eine Strukturierung dieser Methoden, sowie die Toolunterstützung der Kreativprozesse bei der Serviceentwicklung. Ziel dieser der Veranstaltung ist die Entstehung eines ersten Entwurfs für kreative Serviceentwicklungsmethoden, sowie eine Designidee für ein Service-Entwicklungswerkzeug.

Angestrebte Lernergebnisse

Kenntnisse:

Die Studierenden erlernen und erwerben Kenntnisse über

- Modelle der Unternehmens- und Servicestrategie
- Unterschiedliche Methoden und Herangehensweisen bei der Serviceentwicklung
- Neue technologische Entwicklung, deren Möglichkeiten und Einfluss auf die Gestaltung von Services
- Beschreibungsmöglichkeiten für die Modellierung von Serviceentwicklungsmethoden und funktionaler Modellierung und dafür nutzbare Tools

Fertigkeiten:

Die Studierenden werden in die Lage versetzt, selbständig neue Services auf verschiedenen Ebenen im Unternehmen zu entwerfen, zu beschreiben und zu modellieren. Sie erlernen den Umgang mit Tools wie Adonis CE und wenden dies erfolgreich in praktischen Beispielen zur Modellierung innovativer Services an.

Kompetenzen:

Die Studierenden erwerben die Kompetenz neuartige Entwicklungsmethoden für Serviceinnovation zu erdenken und voranzubringen, um kreativ und systematisch neue Services zu entwerfen und im Unternehmensumfeld zu bewerten. Sie haben das das Fachwissen Tools einzuschätzen und auszuwählen, um Serviceinnovationen methodisch wie auch funktional akkurat zu modellieren.

Inhalt

- Einführung und Überblick
- Grundlagen Unternehmensstrategie/Geschäftsmodelle/Servicestrategie
 - Porters Value Chain und 5-forces-Modell
 - INSEAD Blue Ocean Strategy
 - Service Business Model Canvas von Böhmann
- Einführung in das Services-System-Modell von Stephen Alter
 - Service Architecture
 - Value Creation Process
 - Service Life Cycle
- Auswirkung des Technologischen Wandels auf die Gestaltungsoptionen von Services
 - Cloud Computing
 - Ubiquitous Computing
 - Big Data
 - Cyber Physical Systems
 - Web2.0
- Modellierung von Service-Entwicklungsmethoden mit Adonis CE
- Funktionale Modellierung geeigneter Unterstützungstools Adonis CE

Medienformen

PDF der Folien aus der Vorlesung und Beispiele in Fallstudien
Weiteres Material wird während der Vorlesung bekannt gegeben.

Literatur

- Osterwalder, A. & Pigneur, Y. (2010): Business Model Generation: A handbook for visionaries, game changers, and challengers. John Wiley & Sons 2010
- Osterwalder, A. et al. (2014): Value proposition design: How to create products and services customers want. John Wiley & Sons 2014

- Josey, A. et. Al. (2014): Archimate 2.1. A pocket guide. The Open Group Publications. Van Haren Publishing, Zaltbommel NL 2014.
- Sapir, J., Fingar, P. (2014): Master your untamed business processes: How to build smart process applications on the Salesforce1 platform. E-Book, salesforce.com

Modul:	Elective: Internet of Things (IoT)
Kürzel:	DBE21, DBE31
Lehrveranstaltung:	Internet of Things (IoT)
Veranstaltungsformat:	Vorlesung mit integrierter Übung
Studiensemester:	Wintersemester
Modulverantwortlicher:	Prof. Dr. Christian Decker
Dozent(in):	Prof. Dr. Christian Decker
Sprache:	Deutsch
Zuordnung zum Curriculum:	Digital Business Engineering, Wahlfach, 2. und 3.Semester
Lehrform / SWS:	Vorlesung & Übung 4 SWS
Arbeitsaufwand:	Präsenzstudium: 60 Stunden Eigenstudium: 120 Stunden
Kreditpunkte:	6 ECTS
Voraussetzungen nach StuPro:	
Empfohlene Voraussetzungen:	Gute Betriebswirtschafts und Informatikkenntnisse; Kenntnisse im wissenschaftlichen Arbeiten
Studien-/ Prüfungsleistungen/ Prüfungsform:	Vorlesung mit integrierten Übungen; Prüfungsleistung (siehe StuPro) ist eine Projektarbeit, das Thema gibt der Dozent in der Vorlesung bekannt

Modulziele

Das Internet der Dinge, engl. Internet of Things (IoT), beschäftigt sich mit der Informationsverarbeitung in Umgebungen, in denen extrem viele miniaturisierte Rechnersysteme miteinander vernetzt sind und mit Benutzern auf vielfältige Weise interagieren können.

Ziel des Moduls ist es, die Studierenden in die Grundlagen, Technologien und Anwendungsmöglichkeiten des Internet of Things (IoT) einzuführen. Das umfasst ein schichtenübergreifendes Know-How über den Aufbau, Funktionsweise und Vernetzung von Rechnersystemen und deren verteilte Informationsverarbeitung. Dies wird durch die Vermittlung von Wissen in den Bereichen Hardware, Software, Kommunikationsprotokolle, Middleware und Systemdesign erreicht.

Angestrebte Lernergebnisse

Kenntnisse:

- Veränderte Ausprägung der Informationsverarbeitung durch miniaturisierte vernetzte Rechnersysteme

- Wissen über die Technologieanforderungen an Rechnersysteme, die in die reale Welt quasi unsichtbar eingebettet sind
- Kommunikationstechnologien und –protokolle zur massiven Vernetzung von eingebetteten Rechnersystemen
- Möglichkeiten und Einsatz von Sensorik
- Klassifikation von IoT Anwendungen und Entwicklungsmethoden
- IoT Systemdesign, Plattformen und Kommunikationsmustern integrierender Systeme
- Value Driver und Veränderungen von Geschäftsmodellen durch IoT
- Web als Middleware im Web-of-Things (WoT)

Fertigkeiten:

Die Studierenden werden in die Lage versetzt, selbständig auf verschiedenen Ebenen im Unternehmen IoT Anwendungen zu entwerfen und zu entwickeln. Sie entwickeln ein schichtenübergreifendes Verständnis von Rechnersystemen und deren vernetzte Informationsverarbeitung im Zusammenspiel mit neuen Möglichkeiten der impliziten Benutzerinteraktion. Dazu gehört die Fertigkeit zugehörige Managementfunktionen ausüben und IoT Ansätze erfolgreich in Unternehmensanwendungen zu integrieren.

Kompetenzen:

Die Studierenden sind in der Lage, IoT Lösungen zu entwerfen. Durch ein schichtenübergreifendes Verständnis von Rechnersystemen haben sie die Kompetenz die Schlüsseleigenschaften von IoT Technologien einzuschätzen, um neuartige oder verbesserte Anwendungen durch die massive Vernetzung von eingebetteter Informationstechnologie zu verwirklichen.

Inhalt

Das Modul vermittelt die Grundlagen und Konzepte des Themenfeldes „Internet der Dinge“. Es werden Hardware- und Softwaretechnologien, insbesondere zur sensorischen Erfassung und Kommunikationsprotokolle, besprochen. Schwerpunkte bilden die Themenbereiche Smart Object Computer, IoT Plattformen, Anwendungen und Entwicklungsmethoden sowie das Web of Things. Kleinere Aufgaben während der Vorlesung vertiefen die Inhalte. Das Modul behandelt folgende Themenbereiche:

- Einführung und Einordnung in die Entwicklung der Computertechnologie
- Enabling Technologie, Einbettung „The invisible computer“, Smart Object Computer
- Kommunikationsformen von IoT Technologien und sensorische Erfassung
- IoT Anwendungen und Entwicklungsmethoden
- IoT Geschäftsmodelle
- IoT Plattformen für die Integration mit weiteren informationsverarbeitenden Systemen
- Web of Things (WoT)

Medienformen

PDF der Folien aus der Vorlesung. Weiteres Material wird während der Vorlesung bekannt gegeben.

Literatur

- Weiser, M. The computer for the 21st century
- Mattern F., Flörkemeier, Ch. Vom Internet der Computer zum Internet der Dinge. Informatik Spektrum, Vol. 33, no. 2, S. 107-121, April 2010
- Porter, M.E., Heppelmann, J.E., How Smart, Connected Products Are Transforming Competition. Harvard Business Review 92, no. 11, S. 64-88, November 2014

Modul:	Elective: Scientific Publishing	
Kürzel:	DBE21, DBE31	
Lehrveranstaltung:	Scientific Publishing	
Veranstaltungsformat:	Vorlesung und Übungen	
Studiensemester:	Sommer- und Wintersemester	
Modulverantwortlicher:	Prof. Dr. Alfred Zimmermann	
Dozent(in):	Prof. Dr. Alfred Zimmermann	
Sprache:	Deutsch, mit englischen Materialien	
Zuordnung zum Curriculum:	Digital Business Engineering, Wahlfach, 2. und 3. Semester	
Lehrform/ SWS:	Vorlesung & Übung	4 SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	120 Stunden
Kreditpunkte:	6 ECTS	
Voraussetzungen nach StuPro:		
Empfohlene Voraussetzungen:	Grundlagen über wissenschaftliches Arbeiten, Zwischenergebnisse aus Projekten	
Studien-/ Prüfungsleistungen/ Prüfungsform:	Vorlesung mit integrierten Übungen; Bewertete Hausarbeit für: wissenschaftlichen Aufsatz, Poster, Präsentation	

Modulziele

Wissenschaftliche Publikationen sind für Studierende der Schlüssel für bessere Inhalte und deren Kommunikation in Wissenschaft und Praxis und fördern auch den persönlichen Erfolg. Erfahrungen mit eigenen Publikationen qualifizieren Studierende hervorragend für eine weiterführende Rolle in Wissenschaft und Praxis sowie für die Erstellung von erfolversprechenden Projektdokumentationen und von qualitativ hochwertigen Abschlussarbeiten. Studierende erwerben innerhalb des Moduls Scientific Publishing praktische Erfahrungen bei der Vorbereitung eigener Beiträge für wissenschaftliche Konferenzen und lernen Publikationsprozesse exemplarisch kennen. Studierende entwickeln - gestützt durch die Vorlesung und ein individualisiertes Coaching - professionell konzipierte Publikationen im eigenen Forschungsfeld, als Basis einer Einreichung für eine wissenschaftliche Konferenz nach Wahl im kombinierten Profil Wirtschaftsinformatik, Informatik, Wirtschaft. Das Modul vermittelt ebenfalls eine Orientierung über geeignete wissenschaftliche Konferenzen, Journals, und Verlagspublikationen als Basis für erfolversprechende Einreichungen. Dabei sammeln Studierende umfangreiche Kompetenzen durch Vorbereitung eines wissenschaftlichen Aufsatzes, eines Posters sowie einer zugehörigen Konferenzpräsentation.

Angestrebte Lernergebnisse

Kenntnisse:

Im Rahmen des Moduls werden Fachkenntnisse aus ausgewählten Themengebieten des Masterstudiums und der Praxis vertieft. Darüber hinaus erlangen Studierende methodische und praktische Kenntnisse für die Entwicklung eines wissenschaftlichen Aufsatzes und Posters und für deren Einreichung bei wissenschaftlichen Konferenzen. Studierende erwerben darüber hinaus wesentliche Qualitätsregeln guter wissenschaftlicher Praxis und kennen die Reflexionsregeln für Peer-Reviews.

Fertigkeiten:

Studierende erlangen die Fähigkeit zur Formulierung eigener wissenschaftlicher Fragestellungen, der Umsetzung und Erarbeitung eines wissenschaftlichen Aufsatzes und Posters samt Präsentation, sowie der Einreichung eines wissenschaftlichen Beitrages bei Konferenzen. Damit werden grundlegende Fertigkeiten für das wissenschaftliche Arbeiten vertieft und ausgebaut.

Kompetenzen:

Studierende erwerben die Kompetenz zur Analyse, Konzeption und Synthese eigener englischsprachiger Beiträge für wissenschaftliche Konferenzen oder wissenschaftliche Journale. Dabei können Studierende relevante Themenstränge und Forschungsfragen ableiten und eigene Forschungsinteressen darauf beziehen. Schließlich erlangen die Studierenden die Kompetenz für das Management eines eigenen Publikationsprozesses und erwerben Grundkenntnisse über internationale Konferenzen und wissenschaftliche Publikationsplattformen.

Inhalt

Im Rahmen des Moduls werden aktuelle Konferenzformate in der Informatik- und Managementforschung vorgestellt. Auf dieser Basis entwickeln Studierende eigene Forschungsansätze und wissenschaftliche Publikationen. Folgende Themen werden gemeinsam durch Vorlesungen vermittelt und durch angeleitete Teams bearbeitet:

- Motivation und Ziele bei wissenschaftlichen Publikationen und Konferenzen, Journals und Büchern.
- Wissenschaftliche Konferenzen und Wissenschaftsorganisationen in Informatik und Management.
- Publikationsprozesse und Werkzeuge am Beispiel einer Student Conference.
- Analyse von Konferenzaufrufen, Konferenzthemen und Abbildung auf eigene relevante Publikationsideen.
- Konzeption und Synthese relevanter Publikationsthemen und Forschungsfragen mit Bezug auf das eigene Forschungsgebiet.
- Konzeption, Entwicklung und Einreichung eines wissenschaftlichen Aufsatzes, Präsentation in englischer Sprache sowie ein zugehöriges Poster: Formate, Strukturen, Inhalte.
- Qualitätskriterien und Review-Prozesse sowie Regeln guter wissenschaftlicher Praxis.
- Berücksichtigung von Zielformaten und Deadlines bei Konferenzen.
- Unterschiedliche Forschungsmethoden mit Bezug auf das eigene Forschungsthema.
- Projektmanagement und Qualitätsmanagement bei erweiterten wissenschaftlichen Publikationen am Beispiel Scientific Proposal für Master-Thesen und/oder Dissertationen.

Medienformen

Vorlesung, Einzel- und Gruppenarbeit, Coaching von parallelen Teams

Literatur

- Turabian, K.L. (2007): A Manual for Writers of Research Papers, Theses, and Dissertations. University of Chicago Press
- Hacker, D., Sommers, N. (2011): A Writer's Reference. Bedford/St. Martin's
- Wallwork, A. (2011): English for Writing Research Papers. Springer
- Wallwork, A. (2010): English for Presentations at International Conferences. Springer
- O'Leary, Z. (2009): The Essential Guide to Doing Your Research Project. Sage Publications
- King, R.S. (2012): Research Methods for Information Systems, Transatlantic Publishers

Modul:	Elective: Software Process Engineering
Kürzel:	DBE21, DBE31
Lehrveranstaltung:	Software Process Engineering
Veranstaltungsformat:	Vorlesung und Übung/Projekt
Studiensemester:	Wintersemester
Modulverantwortlicher:	Prof. Dr. Marco Kuhrmann
Dozent(in):	Prof. Dr. Marco Kuhrmann
Sprache:	Deutsch
Zuordnung zum Curriculum:	Digital Business Engineering, Wahlfach, 2. und 3. Semester
Lehrform/ SWS:	Vorlesung & Übung, 4 SWS
Arbeitsaufwand:	Präsenzstudium 60 Stunden Eigenstudium 120 Stunden
Kreditpunkte:	6 ECTS
Voraussetzungen nach StuPro:	
Empfohlene Voraussetzungen:	Interesse und Motivation, Prozesse für softwareintensive Produkte erfolgreich zu gestalten
Studien-/ Prüfungsleistungen/ Prüfungsform:	Vorlesung mit integrierten Übungen, Projektarbeit. Die Prüfung umfasst ein Projektportfolio, bestehend aus mehreren Teilleistungen, die innerhalb der Vorlesungszeit erbracht werden.

Modulziele

Die Studierenden können grundlegende Vorgehensmodelle und ihre Rolle in Organisationen und Projekten verstehen. Auf der Grundlage eines systematischen Vorgehens, sind die Teilnehmer nach dem Kurs in der Lage, organisations- und projektspezifische Vorgehensmodelle zu entwickeln, diese zu etablieren und deren kontinuierliche Verbesserung zu begleiten. Zentrale Bestandteile, welche die Teilnehmer nach dem Kurs kennen und anwenden können, sind Metriken/Messungen, Prozessanalysen, sowie Fertigkeiten hinsichtlich der Modellierung und Dokumentation von Vorgehensmodellen.

Angestrebte Lernergebnisse

Kenntnisse:

Die Studierenden erlernen die Grundlagen des Softwareentwicklungsprozesses, die unterschiedlichen Basisprozesse und agilen Methoden, die Konzepte hinter dem Softwareentwicklungsprozess und die Methoden zur Entwicklung und Einführung organisationsweiter und projektspezifischer Entwicklungsprozesse. Auf der Basis eines Lifecycle-Modells erlernen die Studierenden die Analyse von Prozessanforderungen, das Design von Prozessmodellen, Techniken zur Evaluation der Prozessperformanz und die Entwicklung von prozessverbessernden Maßnahmen.

Fähigkeiten:

Die in der Vorlesung vermittelten Kenntnisse über die Methoden und Techniken des Prozessentwurfs werden in den Übungen in Einzel- und Kleingruppenaufgaben vertieft, die sowohl theoretische als auch praktische Aufgaben enthalten.

Kompetenzen:

Die Studierenden sind in der Lage, Methoden und Praktiken für spezifische Projektsituationen auszuwählen und diese in projektspezifischen hybriden Projektprozessen zu kombinieren. Weiterhin verstehen die Studierenden Unternehmensanforderungen an die Prozesse und sind in der Lage, Konzepte des Qualitätsmanagements anzuwenden, um unternehmensweite Prozessstandards einzuführen, Prozessperformanz zu messen und Prozessverbesserungsprojekte durchzuführen. Die Studierenden sind in der Lage, Prozesse zu analysieren, individualisierte Prozesse zu entwerfen und Vorschläge zur Prozessverbesserung zu entwickeln. Weiterhin sind die Studierenden in der Lage, einfache Messinstrumente zur Ermittlung der Prozessperformanz und zur Analyse der Stärken und Schwächen zu entwickeln und anzuwenden.

Inhalt

Dieser Kurs vermittelt Grundlagenwissen, sowie fortgeschrittene Konzepte, Methoden und Techniken des Software Process Engineering, insbesondere:

- Grundlegende Vorgehensmodelle, z.B. Phase/Stage-Gate Modelle, Wasserfall, Spiralmodell
- Agile und Lean Software Development Methods, z.B. Scrum, Kanban
- Agile Scaling Models, z.B. SAFe, Less
- Hybrid Development Methods, z.B. der „Water-Scrum-Fall“
- Process Quality and Maturity Models, z.B. CMMI, ISO 15504
- Software Process Lines
- Software Process Improvement (SPI)
- Software Process Metamodels
- Techniken zur Anpassung von Vorgehensmodellen an Organisationen und Projekte
- Fortgeschrittene Techniken für Analyse, Entwurf, Realisierung und die gesteuerte Evolution von Vorgehensmodellen
- Qualitätsmanagement-Konzepte und Methoden zur Messung von Projektleistungs- und zur Verbesserung organisationsweiter und projektspezifischer Vorgehensmodelle

Medienformen

Präsentation mit Beamer und Tafel, Übungsaufgaben, Projektaufgaben

Literatur

- Münch, Armbrust, Kowalczyk, Soto: Software Process Definition and Management. Springer, 2012.
- Kuhrmann, Münch, Richardson, Rausch, Zhang: Managing Software Process Evolution. Springer, 2016
- Kneuper: Software Processes and Life Cycle Models: An Introduction to Modelling, Using and Managing Agile, Plan-Driven and Hybrid Processes. Springer, 2018

Modul:	Elective: Agile Project Management and Software Development
Kürzel:	DBE21, DBE31
Lehrveranstaltung:	Agile Project Management and Software Development (APM)
Veranstaltungsformat:	Vorlesung und Übung/Projekt
Studiensemester:	Sommersemester
Modulverantwortlicher:	Prof. Dr. Marco Kuhrmann
Dozent(in):	Prof. Dr. Marco Kuhrmann
Sprache:	Deutsch
Zuordnung zum Curriculum:	Digital Business Engineering, Wahlfach, 2. und 3. Semester
Lehrform/ SWS:	Vorlesung & Übung, 4 SWS
Arbeitsaufwand:	Präsenzstudium 60 Stunden Eigenstudium 120 Stunden
Kreditpunkte:	6 ECTS
Voraussetzungen nach StuPro:	
Empfohlene Voraussetzungen:	Interesse und Motivation, Agile Projekte und Softwareentwicklungsprozesse erfolgreiche durchzuführen
Studien-/ Prüfungsleistungen/ Prüfungsform:	Vorlesung mit integrierten Übungen, Hausarbeit Die Prüfung umfasst eine Hausarbeit (schriftliche Ausarbeitung). Die Hausarbeit wird im Rahmen einer Präsentation im Plenum vorgestellt

Modulziele

Die Studierenden können grundlegende Techniken und Methoden des Projektmanagements im Kontext der agilen Softwareentwicklung. Die Teilnehmer sind nach dem Kurs in der Lage, grundlegende Techniken, Methoden und Praktiken des Projektmanagements in agilen Softwareentwicklungsprojekten anzuwenden, z.B. Kostenschätzungs- und Planungstechniken. Ferner sind die Teilnehmer in der Lage, Teams zu organisieren und gruppendynamische Effekte in der Teamarbeit zu verstehen.

Angestrebte Lernergebnisse

Kenntnisse:

Die Studierenden erlernen die Grundlagen des Projektmanagements im Kontext der agilen Softwareentwicklung. Sie lernen unterschiedliche agile Methoden. Auf der Basis eines Lifecycle-Modells erlernen die Teilnehmer Techniken, Methoden und Praktiken zur Schätzung, Planung, Anforderungserhebung, Codierung und Qualitätssicherung, sowie der Vertragsgestaltung. Die Teilnehmer lernen Modelle kennen, welche helfen, gruppensdynamische Effekte zu verstehen und kritische Situationen in Projekten zu beherrschen.

Fähigkeiten:

Die in der Vorlesung vermittelten Kenntnisse über die Techniken, Methoden und Praktiken des Projektmanagements und der Softwareentwicklung werden in den Übungen in Einzel- und Kleingruppenaufgaben vertieft, die sowohl theoretische als auch praktische Aufgaben enthalten.

Kompetenzen:

Die Studierenden sind in der Lage, Techniken, Methoden und Praktiken des Managements und der Entwicklung in agilen Projekten zielorientiert auszuwählen und einzusetzen. Weiterhin verstehen die Studierenden wie es in Projekten zu kritischen Situationen kommt und wie man in solchen kritischen Situationen reagieren kann, um sie zu lösen. Dazu werden u.a. empirische Methoden erlernt, um den Projektstatus und die Projektperformance messen zu können.

Inhalt

Dieser Kurs vermittelt Grundlagenwissen, sowie fortgeschrittene Techniken, Methoden und Praktiken des Projektmanagement und der Softwareentwicklung, insbesondere:

- Agile und Lean Methoden, z.B. Scrum, Kanban
- Agile Skalierungsmodelle, z.B. SAFe, Less
- Techniken zur Kostenschätzung und Projektplanung
- Techniken zur Anforderungserhebung/zum Anforderungsmanagement
- Codierungstechniken
- Verteilte Softwareentwicklung

Medienformen

Präsentation mit Beamer und Tafel, Übungsaufgaben, Projektaufgaben

Literatur

- Schwaber: Agile Project Management with Scrum. Microsoft Professional, 2004.
- Highsmith: Agile Project Management. Addison-Wesley, 2010
- Martin: Clean Architecture. Prentice Hall, 2018
- Martin: Clean Code, Prentice Hall, 2009
- Broy, Kuhmann: Einführung in die Softwaretechnik, Springer, 2021

Modul:	Elective: Software Integration	
Kürzel:	DBE21, DBE31	
Lehrveranstaltung:	Software Integration	
Veranstaltungsformat:	Vorlesung und Übungen	
Studiensemester:	Wintersemester und Sommersemester	
Modulverantwortlicher:	Prof. Dr. Uwe Breitenbücher	
Dozent(in):	Prof. Dr. Uwe Breitenbücher	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Digital Business Engineering, Wahlfach, 2. und 3. Semester	
Lehrform/ SWS:	Vorlesung	2 SWS
	Übungen	2 SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	120 Stunden
Kreditpunkte:	6 ECTS	
Anteil Informatik/Wirtschaftswiss.	80% / 20%	
Voraussetzungen nach StuPro:		
Empfohlene Voraussetzungen:	Grundlegendes Wissen im Bereich Softwareentwicklung. Grundlegende Programmierkenntnisse in einer der folgenden Programmiersprachen sind ausreichend: Java, Python, PHP, JavaScript, C#, Ruby	
Studien-/ Prüfungsleistungen/ Prüfungsform:	Vorlesung mit integrierten Übungen; Prüfung in Form einer schriftlichen Klausur (60min)	

Modulziele

IT-Systeme bestehen häufig nicht nur aus einem einzelnen Softwareprogramm, sondern aus mehreren unabhängig voneinander entwickelten Softwarekomponenten, die miteinander integriert werden müssen, um die Gesamtfunktionalität eines Systems zu realisieren. Ziel dieses Moduls ist es, die Studierenden in die Grundlagen, Konzepte und Technologien der Softwareintegration einzuführen. Die Studierenden lernen dazu unterschiedliche Architekturstile kennen, die zur Integration von heterogenen Softwarekomponenten eingesetzt werden können, und können deren jeweilige Eignung für einen bestimmten Anwendungsfall beurteilen. Die Teilnehmer sind nach dem Besuch dieses Kurses in der Lage, Architekturen für umfangreiche Systeme zu entwerfen, in denen zahlreiche Softwarekomponenten zusammenarbeiten müssen. Des Weiteren

können die entworfenen Systemarchitekturen von den Studierenden in der Praxis auf Basis etablierter Technologien umgesetzt und implementiert werden.

Angestrebte Lernergebnisse

Kenntnisse:

Die Studierenden lernen die umfangreichen Probleme der Softwareintegration kennen und verstehen die Prinzipien der engen und losen Kopplung von Softwarekomponenten. Durch die Einführung und Diskussion unterschiedlicher Architekturstile wie Microservices, REST und Pipes-and-Filters sowie zahlreicher Patterns erwerben die Teilnehmer detaillierte Kenntnisse darüber, welche Ansätze sich für die Integration von Softwarekomponenten in der Praxis bewährt haben und welche Konsequenzen die jeweiligen Ansätze für die Weiterentwicklung, Wartbarkeit und Anpassbarkeit des Gesamtsystems haben. Ein Schwerpunkt dieser Lehrveranstaltung liegt dabei auf der Messaging-Technologie, wodurch die Studierenden lernen, wie Softwarekomponenten asynchron miteinander kommunizieren können und welche Vor- und Nachteile diese Kommunikationsform gegenüber synchronen Integrationslösungen hat.

Fertigkeiten:

Die Studierenden erlangen die Fähigkeit, selbständig Systemarchitekturen für die Integration heterogener Softwarekomponenten zu entwerfen und können diese in der Praxis mit etablierten Technologien umsetzen. Sie entwickeln ein Verständnis über das Zusammenspiel von Architekturstilen, Patterns und Technologien und können dadurch konkrete Integrationsprobleme abstrahieren, konzeptionell lösen und die entwickelten Lösungen anschließend technisch realisieren. Diese Fertigkeiten werden in den zugehörigen Übungen in Form von Einzel- und Kleingruppenaufgaben vertieft.

Kompetenzen:

Die Studierenden erwerben die Kompetenz, geeignete Systemarchitekturen für Integrationsprobleme zu entwickeln und diese praktisch umzusetzen. Weiterhin können die Studierenden nach dem Besuch des Kurses sowohl Architekturen als auch deren Umsetzung hinsichtlich unterschiedlicher nicht-funktionaler Eigenschaften wie Anpassbarkeit und Skalierbarkeit analysieren, beurteilen und optimieren.

Inhalt

- Herausforderungen bei der Integration von heterogenen Softwarekomponenten
- Grundlegende Integrationsansätze, z.B. Shared Database
- Client-Server Architekturstil und Remote Procedure Calls
- Unterschiedliche Architekturstile für die Softwareintegration, z.B. REST und Pipes-and-Filters
- Prinzipien der engen und losen Kopplung von Softwarekomponenten
- Messaging-Technologie und moderne Messaging Middlewares am Beispiel RabbitMQ
- Integration von Softwarekomponenten auf Basis der Messaging-Technologie
- Integration proprietärer Anwendungen
- Architekturmuster für die Entwicklung Message-basierter Integrationslösungen

Medienformen

Präsentation mit Beamer und Tafel sowie Übungsaufgaben.

Literatur

- G. Hohpe and B. Woolf: "Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions." Addison-Wesley Professional, ISBN-13: 978-0321200686, 2003.
- Lovisa Johansson and David Dossot: "RabbitMQ Essentials: Build distributed and scalable applications with message queuing using RabbitMQ." Packt Publishing, ISBN-13: 978-1789131666, 2020.

Modul:	Elective: Applied Machine Learning	
Kürzel:	DBE21, DBE31	
Lehrveranstaltung:	Applied Machine Learning	
Veranstaltungsformat:	Vorlesung und Übung / Projekt	
Studiensemester:	Sommersemester	
Modulverantwortlicher:	Prof. Dr. Michael Möhring	
Dozent(in):	Prof. Dr. Michael Möhring, Prof. Dr. Marco Kuhmann, Prof. Dr. Christian Decker	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Digital Business Engineering, Wahlfach, 2. und 3. Semester	
Lehrform/ SWS:	Vorlesung mit integrierten Übungen 4 SWS	
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	120 Stunden
Kreditpunkte:	6 ECTS	
Anteil Informatik/Wirtschaftswiss.	80% / 20%	
Voraussetzungen nach StuPro:	Da Modul ist auf max. 15 Teilnehmende beschränkt.	
Empfohlene Voraussetzungen:		
Studien-/ Prüfungsleistungen/ Prüfungsform:	Vorlesung mit integrierten Übungen; Prüfung in Form Projektarbeit	

Modulziele

Die Studierenden kennen die grundlegenden Begriffe und Verfahren zum Entwurf und zur Entwicklung von Anwendungen, welche KI-/ML-Komponenten enthalten. Sie können zu gegebenen Problemen Lösungen entwickeln und diese umsetzen. Ferner sind die Studierenden in der Lage, KI-/ML-Systeme nach unterschiedlichen Gesichtspunkten zu bewerten, insbesondere sinnvolle Anwendung, Implikationen der Anwendung, wie ethische und regulatorische Aspekte der KI-/ML-Technologien.

Angestrebte Lernergebnisse

Kenntnisse:

Die Studierenden erlernen die Grundlagen im Themengebiet KI/ML. Sie können Ansätze für unstrukturierte sowie (semi-) strukturierte Daten und für deren Kombination unterscheiden sowie anwenden. Sie lernen unterschiedliche Methoden kennen, mit denen sich KI-/ML-Systeme konzipieren, entwerfen und umsetzen lassen. Dies umfasst

neben dem Requirements Engineering den Entwurf/die Architektur und auch die Umsetzung/Anwendung von KI-/ML-Systemen. Die Bewertung von KI-/ML-Systemen nach unterschiedlichen Gesichtspunkten wird erlernt und praktiziert.

Fertigkeiten:

Die in der Vorlesung vermittelten Kenntnisse über die Techniken, Methoden und Praktiken des Entwurfs und der Entwicklung von KI-/ML-Systemen werden in den Übungen in Einzel- und Kleingruppenaufgaben vertieft, die sowohl theoretische als auch praktische Aufgaben enthalten.

Kompetenzen:

Die Studierenden sind in der Lage, Techniken, Methoden und Praktiken des Entwurfs und der Entwicklung von KI-/ML-Systemen zielorientiert auszuwählen und einzusetzen. Weiterhin verstehen die Studierenden, welche spezifischen Herausforderungen bei der Anwendung von KI-/ML-Techniken zu meistern sind. Grenzen der jeweiligen Methode können mittels etablierter Evaluationstechniken dargestellt und für den jeweiligen Anwendungsfall diskutiert werden. Dazu werden u.a. mathematische, empirische Methoden erlernt, um die theoretische Grundlage zu legen.

Inhalt

Dieser Kurs behandelt insb. die folgenden Themen, je nach Themensetzung:

- Theoretische Grundlagen KI/ML im Allgemeinen
- Analysemethoden für strukturierte, unstrukturierte und semi-strukturierte Daten (bspw. Regression, Entscheidungsbäume, Künstliche Neuronale Netze / Deep Learning, Text Mining / NLP, Image Object Recognition, Process Mining, etc.)
- Anforderungserhebung (Requirements Engineering) für KI/ML
- Entwurf (Architektur) von KI-/ML-Systemen
- KI-/ML-Frameworks
- Datenbanken (SQL/No-SQL) zur Speicherung und Zugriff auf Daten verschiedener Struktur sowie Aspekte zur Datenqualität
- DevOps, CI/CD und Testing von KI/ML Systemen
- Ethische und regulatorische Aspekte von KI/ML

Hinweis: Das Modul ist auf maximal 15 Teilnehmende beschränkt.

Medienformen

Präsentation mit Beamer und Tafel, Übungsaufgaben, Projektaufgaben

Literatur

- Kotu, Vijay, and Bala Deshpande. Predictive analytics and data mining. Morgan Kaufmann, 2014.
- Tan, Ah-Hwee. Text mining: The state of the art and the challenges. In Proceedings of the pakdd 1999 workshop on knowledge discovery from advanced databases, vol. 8, pp. 65-70. 1999.
- Van der Plas, Jake. Python data science handbook: Essential tools for working with data. " O'Reilly Media, Inc.", 2016.
- Van Der Aalst, Wil. "Process mining: Overview and opportunities." ACM Transactions on Management Information Systems (TMIS) 3, no. 2 (2012): 1-17.
- Norvig, R. Artificial Intelligence – a modern approach. 3rd edition, Pearson

Modul:	Elective: Cloud-based Web Application Development	
Kürzel:	DBE21, DBE31	
Lehrveranstaltung:	Cloud-based Web Application Development	
Veranstaltungsformat:	Vorlesung und Übungen	
Studiensemester:	Sommersemester	
Modulverantwortlicher:	Prof. Dr. Uwe Breitenbücher	
Dozent(in):	Prof. Dr. Uwe Breitenbücher	
Sprache:	Deutsch	
Zuordnung zum Curriculum:	Digital Business Engineering, Wahlfach, 2. und 3. Semester	
Lehrform/ SWS:	Vorlesung	2 SWS
	Übungen	2 SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	120 Stunden
Kreditpunkte:	6 ECTS	
Anteil Informatik/Wirtschaftswiss.	80% / 20%	
Voraussetzungen nach StuPro:		
Empfohlene Voraussetzungen:	Grundlegendes Wissen im Bereich Softwareentwicklung.	
Studien-/ Prüfungsleistungen/ Prüfungsform:	Vorlesung mit integrierten Übungen; Projektarbeit	

Modulziele

Software wird immer häufiger in Form von Web-Anwendungen entwickelt, die in der Cloud betrieben werden und dem Nutzer einen schnellen Zugang via Browser ohne zusätzliche Softwareinstallationen ermöglichen. Ziel dieses Moduls ist es daher, die Studierenden in die Konzepte, Architekturen und Technologien der Entwicklung von Cloud-basierten Web-Anwendungen einzuführen. Die Studierenden lernen dazu die Grundlagen der Web-Programmierung und des Cloud Computings kennen und befassen sich mit der Anwendung unterschiedlicher Architekturstile, um Web-Anwendungen so zu entwickeln, dass diese elastisch skalierbar in der Cloud große Nutzerzahlen bewältigen können. Die Teilnehmer sind nach dem Besuch des Moduls in der Lage, gegebene Systemarchitekturen zu analysieren und deren Eignung für einen Betrieb in der Cloud zu bewerten. Des Weiteren umfasst das Modul alle notwendigen konzeptionellen Grundlagen der Web-Entwicklung, sodass sich die Studierenden anschließend selbständig weiter vertiefen können.

Angestrebte Lernergebnisse

Kenntnisse

Die Studierenden lernen Konzepte und Technologien der Entwicklung von Web-Anwendungen kennen und verstehen die Aufteilung in Frontend und Backend. Sie lernen und üben den Umgang mit HTML, CSS und JavaScript, um Browser-basierte Frontends entwickeln zu können. Der Schwerpunkt dieser Lehrveranstaltung liegt dabei jedoch nur auf den Grundlagen der genannten Sprachen. Der Hauptfokus des Moduls liegt auf den Architekturen und Technologien Cloud-basierter Backends von Web-Anwendungen. Durch die Behandlung unterschiedlicher Architekturstile wie Service-Orientierung, Microservices, REST und Pipes-and-Filters erwerben die Teilnehmer detaillierte Kenntnisse darüber, welche Ansätze sich für große Cloud-basierte Systeme eignen und welche Konsequenzen die jeweiligen Ansätze für die Skalierbarkeit, Elastizität, Wartbarkeit und Anpassbarkeit des Gesamtsystems haben.

Fertigkeiten

Die Studierenden erlangen die Fähigkeit, selbständig (1) einfache Web-Anwendungen zu entwickeln und (2) Architekturen für Cloud-basierte Web-Anwendungen zu entwerfen. Sie entwickeln ein Verständnis über das Zusammenspiel und die Abhängigkeiten von Architekturen und Technologien und können dadurch große Systeme entwerfen und diese prototypisch realisieren. Diese Fertigkeiten werden in den integrierten Übungen in Form von Einzel- und Gruppenaufgaben vertieft.

Kompetenzen

Die Studierenden erwerben die Kompetenz, geeignete Systemarchitekturen für Web-Anwendungen in der Cloud zu entwickeln und diese praktisch umzusetzen. Weiterhin können die Studierenden nach dem Besuch des Kurses sowohl Architekturen als auch deren Umsetzung hinsichtlich unterschiedlicher nicht-funktionaler Eigenschaften wie Anpassbarkeit und Skalierbarkeit analysieren, beurteilen und optimieren.

Inhalt

- Grundlagen der Frontend-Entwicklung mit HTML, CSS und JavaScript
- Grundlagen der Backend-Entwicklung mit NodeJS
- Entwicklung von HTTP APIs
- Single Page Applications, Client-Side Rendering, Server-Side Rendering
- Moderne Architekturen für Web Application Backends, darunter Service-Orientierung, Microservices, REST, etc.
- Cloud-basierte Architekturen für elastisch skalierbare Web-Anwendungen
- Deployment von Web-Anwendungen auf AWS
- Prinzipien der engen und losen Kopplung von Softwarekomponenten

Medienformen

Präsentation mit Beamer und Whiteboard sowie Übungsaufgaben.

Literatur

- Erl, T. Cloud Computing: Concepts, Technology & Architecture, Prentice Hall, 2013. ISBN: 978-0133387520
- Erl, T. und Cope, R. Cloud Computing Design Patterns, Prentice Hall, 2015. ISBN: 978-0133858563
- Fehling, C., Leymann, F. et al. Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications, Springer, 2014. ISBN: 978-3709115671
- Nadareishvili, I, et al.: Microservice Architecture: Aligning Principles, Practices, and Culture, O'Reilly, 2016. ISBN: 978-1491956250

Modul:	Elective: Algorithmics	
Kürzel:	DBE21, DBE31	
Lehrveranstaltung:	Diskrete Optimierung	
Veranstaltungsformat:	Vorlesung und Übung	
Studiensemester:	Wintersemester	
Modulverantwortlicher:	Prof. Dr. Christian Decker	
Dozent(in):	Prof. Dr. Stefan Funke	
Sprache:	Englisch	
Zuordnung zum Curriculum:	Digital Business Engineering, Wahlfach, 2. und 3.Semester	
Lehrform/SWS:	Vorlesung	3 SWS
	Übung	1 SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	120 Stunden
Kreditpunkte:	6 ECTS	
Anteil Informatik/Wirtschaftswiss.	100% / 0%	
Voraussetzungen nach StuPro:		
Empfohlene Voraussetzungen:	Mathematisches Verständnis	
Studien-/Prüfungsleistungen/ Prüfungsform:	Vorlesung findet an der Uni Stuttgart als wöchentliche Veranstaltung statt: Klausur an der Universität Stuttgart (siehe StuPro); Übungsleistung als Prüfungsvoraussetzung	
Besondere Hinweise	Veranstaltung an der Uni Stuttgart; Hinweise zur Durchführung werden von den Dozenten der Uni Stuttgart mitgeteilt und kommen nicht über RELAX.	

Objectives

Students will gain insights and learn methods, techniques and tools for the mathematical discipline of discrete optimization.

Learning outcomes

Skills

Students will achieve an overview on the standard techniques of the area of discrete optimization. They are enabled to approach problems of practical relevance using the tool set from this course and can judge the complexity of those problems.

Knowledge:

Students have the knowledge to model and express optimization problems as integer linear programs. As a consequence, they know how to design approximation algorithms.

Competencies

After the successful completion of this course, students are enabled to recognize optimization problems and judge their complexity.

Content

- Integer linear programming
- Approximation algorithms
- non approximability

Schedule

- Weekly course, regularly taking place twice a week.
- All course dates announced in the calendar on the HHZ website.

Material

Script, assignments for exercise and personal notes

Literature

- David P. Williamson and David B. Shmoys. 2011. *The Design of Approximation Algorithms* (1st ed.). Cambridge University Press, New York, NY, USA.

The course instructor provides a script.

Modul:	Elective: Data Management & Analysis	
Kürzel:	DBE21, DBE31	
Lehrveranstaltung:	Data-Warehouse, Data-Mining und OLAP-Technologien	
Veranstaltungsformat:	Vorlesung und Übungen	
Studiensemester:	Wintersemester	
Modulverantwortlicher:	Prof. Dr. Christian Decker	
Dozent(in):	Prof. Dr. Bernhard Mitschang	
Sprache:	Englisch	
Zuordnung zum Curriculum:	Digital Business Engineering, Wahlfach, 2. und 3.Semester	
Lehrform/ SWS:	Vorlesung	2 SWS
	Übungen	2 SWS
Arbeitsaufwand:	Präsenzstudium	60 Stunden
	Eigenstudium	120 Stunden
Kreditpunkte:	6 ECTS	
Anteil Informatik/Wirtschaftswiss.	80% / 20%	
Voraussetzungen nach StuPro:		
Empfohlene Voraussetzungen:	Grundlegendes Wissen in Datenbanken, ins. Datenmodelle und Modellierung, Relationale Algebra und Anfrageverarbeitung, SQL Grundlagen, Grundlagen in Transaktionsverarbeitung	
Studien-/ Prüfungsleistungen/ Prüfungsform:	Vorlesung findet an der Uni Stuttgart als wöchentliche Veranstaltung statt; Klausur: siehe StuPro; Übungsleistung als Prüfungsvoraussetzung	
Besondere Hinweise	Veranstaltung an der Uni Stuttgart; Hinweise zur Durchführung werden von den Dozenten der Uni Stuttgart mitgeteilt und kommen nicht über RELAX.	

Objectives

The goal of the course is to provide deep understanding of the principles of data management, processing and analytics as well as of how DBMS are implemented and

optimized. The course covers two areas: firstly, Database systems internals, and secondly, data warehouse architectures as well as data analytics.

Learning outcomes

Knowledge:

Database systems internals are addressed in terms of the following topics:

- DBMS Architecture, Storage Media and Hierarchy, DB Engineering Rules of Thumb
- Buffer Management
- Record, page, and file formats and organization, Access Paths and Indexing
- Implementation of relational operators, Query Optimization
- Transaction Processing, Concurrency Control, Recovery

Data warehouse architectures as well as data analytics address the following questions:

- Understand the challenges behind the integration of heterogeneous data sources in consolidated data warehouses and the provisioning of analytical services
- Typical data warehouse architectures and systems as well as current trends
- Structuring of a data warehouse and ETL processes (extraction, transformation, load)
- Data analysis technologies for reporting, online analytic processing and data mining, and their role as part of analytic services

Skills:

The course fosters students ability to select appropriate systems based on specific types of user and systems requirements. It will develop the ability to instrument systems and algorithms based on specific application and hardware settings. In addition it will enable students to analyze data query processing pitfalls and recommend possible performance improvement approaches. Last but not least, it will develop the ability to properly apply various data analysis techniques and implement them in analytical systems.

Competencies:

Students will gain a certain level of proficiency in different areas of data management, data processing and analytics. They will develop understanding of the various types of modern data processing systems architectures and be able to select appropriate ones depending on their advantages as well as on the requirements of the target scenarios. In addition the students will have an understanding of trends in application development, workloads and hardware and derive the impact on data processing and analytics algorithms. Finally, the students will develop deeper understanding of the main tuning options and performance improvement techniques.

Content

The main topics covered throughout the course are:

Database systems internals:

- DBMS Architecture, Storage Media and Hierarchy, DB Engineering Rules of Thumb, Impact of novel hardware
- Buffer Management
- Record, page, and file formats and organization, Access Paths and Indexing
- Implementation of relational operators, Query Optimization
- Transaction Processing, Concurrency Control, Recovery

Data Warehousing and Analytics:

- Introduction to data warehousing and analytical data processing
- Data warehouse architecture and design design

- ETL processes: Extraction, Transformation, Load and ETL as a service
- Introduction to analytics and analytic services
- Online analytic processing and real-time reporting
- Data mining

Schedule

- Weekly course, regularly taking place twice a week.
- All course dates announced in the calendar on the HHZ website.

Material

Course instructor provides slides and script. Course contains practical exercises and assignments.

Literature

- Härder, T., Rahm, E. *Datenbanksysteme - Konzepte und Techniken der Implementierung*. Springer, 2001.
- Ramakrishnan, R., Gehrke, J. *Database Management Systems*. 3rd Ed., McGraw Hill, 2003.
- Garcia-Molina, H., Ullman, J., Widom, J. *Database Systems: The Complete Book*. 2nd Ed., McGraw Hill, 2000.
- Gray, J., Reuter, A. *Transaction Processing: Concepts and Techniques* Morgan Kaufmann, 1993.
- Graefe, G. *Query evaluation techniques for large databases*; ACM Comp. Surv., 25:2, 1993, pp. 73-170.
- V. Köppen, G. Saake, K.-U. Sattler. *Data Warehouse Technologien*, Hüthig Jehle Rehm, 2012.
- W.H. Inmon. *Building the Data Warehouse*. 4th Edition, Wiley & Sons, New York, 2005.
- A. Bauer, H. Günzel. *Data Warehouse Systeme – Architektur, Entwicklung, Anwendung*. 3. Auflage, dpunkt.verlag, Heidelberg, 2008.