

Fakultät Informationstechnik

Modulhandbuch Master-Studiengang **Angewandte Informatik**

Inhaltsverzeichnis

Kern- und Pflichtmodule

Modulnummer	Modul	Seite
1. Semester		
AI 101	Sprachen und Automaten	5
AI 102	Security Engineering	7
AI 103, AI 104	Wahlpflichtmodul 1 und 2	9
AI 105	Forschungsprojekt 1	10
2. Semester		
AI 201	Advanced Software Engineering	12
AI 202	IT Management	14
AI 203, AI 204	Wahlpflichtmodul 3 und 4	16
AI 205	Forschungsprojekt 2	17
3. Semester		
AI 301	Masterarbeit	19
AI 302	Publikation	21

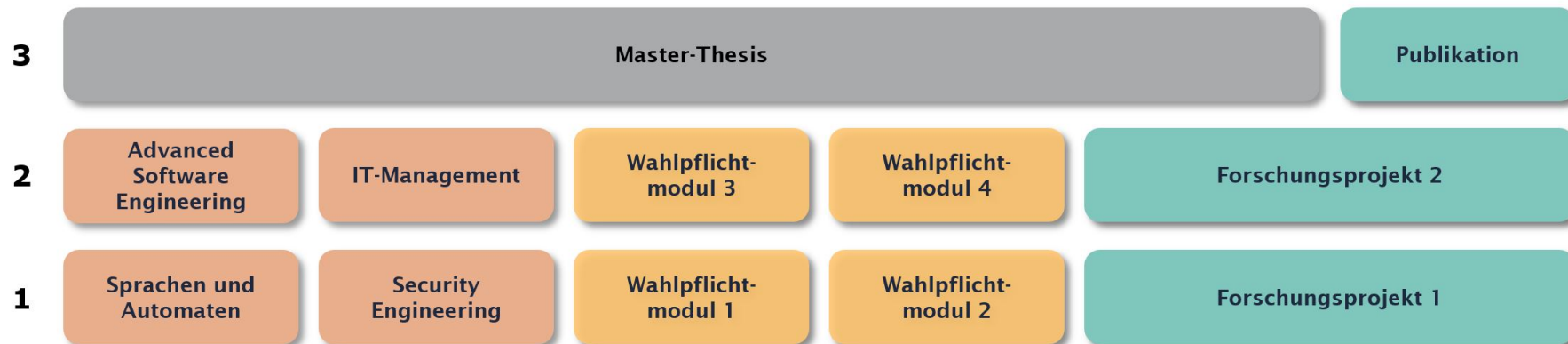
Wahlpflichtmodule und Schwerpunkte

Alle angebotenen Wahlpflichtmodule sind den drei Schwerpunkten Autonome Systeme, Business Intelligence und Mobile Computing zugeordnet. Werden nur Module aus einem Schwerpunkt belegt, so wird dieser auf Antrag im Zeugnis ausgewiesen. Anträge sind zu Beginn des 3. Fachsemesters bei der Fakultät einzureichen.

Die jeweilige Zuordnung der Wahlpflichtmodule zu den Schwerpunkten wird per Aushang durch den Vorsitzenden des Prüfungsausschusses veröffentlicht.

Modulnummer	Modul	Seite
AI W01	Intelligent Data Analytics	23
AI W02	Business Intelligence	25
AI W03	Mobile Communication	27
AI W04	Datenfusion	29
AI W05	Automotive System Design	31
AI W06	Advanced Control	33
AI W12	Web Services	35
AI W14	Data Mining	39
AI W13	Mobile Applikationen	37
AI W15	Mobile UX Design	41
AI W16	Web Technologien	43
AI W18	Network Management	45
AI W19	Cloud Computing	47
AI W20	Advanced Data Models	49
AI W21	High Performance Computing	51
AI W22	Bewegungsplanung für autonome Systeme	53
AI W23	Information Security Management	55
AI W24	Network Security	57
AI W25	Secure Coding	59

Übersicht Modulplan



Legende: Kernmodule Vertiefung Forschung Thesis

Wahlpflichtmodule / Vertiefung



Modulbeschreibung Sprachen und Automaten

Schlüsselworte: Formale Sprachen, Automaten, Turing Maschinen, Komplexitätstheorie

Zielgruppe:	1. Semester AIM	Modulnummer:	AI 101
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Steffen Schober		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Grundlagen der Informatik

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- Grundbegriffe und mathematische Grundlagen der Theoretischen Informatik
- Automaten und Sprachen
 - Endliche Automaten und Reguläre Sprachen
 - Kellerautomaten und Kontextfreie Sprachen
 - Turingmaschinen und rekursiv aufzählbare Sprachen
- Grundlagen der Berechenbarkeit
 - Unentscheidbarkeit
 - Halteproblem
- Grundlagen der Komplexitätstheorie (Klassen P und NP)

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- den Aufbau formaler Sprachen zu beschreiben
- prinzipielle Schranken bestimmter Berechnungsmodelle zu verstehen
- Probleme mit formalen Sprachen und den ihnen zugeordneten Automaten zu lösen

Übergreifende Kompetenzen

Die Studierenden können

- Techniken der Theoretischen Informatik in praktischen Anwendungen nutzen
- kombinatorisch schwerer Berechnungsprobleme erkennen

Literaturhinweise:

Schöning: Theoretische Informatik – kurzgefasst, Spektrum Akademischer Verlag
Hopcroft, u.a.: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Pearson Studium

Wird angeboten:

in jedem Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen, Nachbereitung und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernergebnisse:

Die Studierenden erwerben das notwendige theoretische Wissen, um den Aufbau und Probleme formaler Sprachen und Automaten zu beschreiben. Sie verfügen über die Fähigkeit Techniken der Theoretischen Informatik in praktischen Anwendungen anzuwenden. Sie können damit verschiedene Fragestellungen und Probleme mit formalen Sprachen und den ihnen zugeordneten Automaten lösen und im Hinblick auf ihre Vollständigkeit, Abgeschlossenheit, Berechenbarkeit und Komplexität beurteilen.

Bildung der Modulnote:

Klausur

Modulbeschreibung Security Engineering

Schlüsselworte: Security, Engineering, Software, Design, Standards

Zielgruppe:	1. Semester AIM	Modulnummer:	AI 102
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Englisch		
Modulverantwortung:	Prof. Dr. Dominik Schoop		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Grundlagen des Software Engineerings

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, nach der Einschätzung des Sicherheitsrisikos, Technologien und Vorgehensweisen anzuwenden, um ein sicheres, komplexes und heterogenes IT-System zu entwickeln.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Moderne Verschlüsselungsverfahren
- Kryptographische Protokolle
- Security System Modelling
- Security Requirements Engineering
- Security Design Principles
- Verifikation von Sicherheitskomponenten

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Sicherheitssysteme zu analysieren,
- Sicherheitsrisiken zu analysieren,
- Systeme auf Sicherheit zu validieren und zu verifizieren
- Informationstechnisch sichere Systeme zu entwerfen

Übergreifende Kompetenzen

Die Studierenden können

- nach der Einschätzung des Sicherheitsrisikos, Vorgehensweisen und Technologien anwenden, um ein sicheres, komplexes und heterogenes IT-System zu entwickeln.

Inhalt:

- Einführung in die Kryptographie
- Diskrete Mathematik
- Moderne Verschlüsselungsverfahren
- Kryptographische Protokolle
- Blockchain-Technologie
- Elliptische Kurven
- Beweisbar sichere Kryptographie
- Quantenkryptographie
- Security risk assessment
- Security certification (BSI, CC, EAL)
- Economics in security
- Security standards (ISO, BSI, CIS)

- Security design (security requirements, design criteria, security design patterns, UMLSec)
- Specific security technologies (embedded security, web security, database security)
- Secure coding
- Security testing
- Secure operation

Literaturhinweise:

Ross Anderson: Security Engineering: A Guide to Building Dependable Distributed Systems, Wiley, 2008

Jonathan Katz and Yehuda Lindell: Introduction to Modern Cryptography, Chapman and Hall/CRC, 2014

Bruce Schneier: Angewandte Kryptographie, Pearson Studium, 2006

Wird angeboten:

in jedem Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform: Vorlesung mit Übungen und Security Case Study

Leistungskontrolle: Klausur (90 Minuten), Bericht

Anteil Semesterwochenstunden: 4 SWS

Geschätzte studentische Arbeitszeit: 150 Stunden

Lernergebnisse:

Die Studierenden beherrschen die Methoden zur Einschätzung von Risiken und sind in der Lage, der Situation angemessene Methoden für den sicheren Entwurf, Implementierung und Betrieb eines IT-Systems auszuwählen und anzuwenden. Die Studierenden werden befähigt, die IT-Sicherheit von Anwendungen zu analysieren und zu bewerten. Sie sind in der Lage angemessene IT-Sicherheitslösungen systematisch zu entwickeln und darzustellen.

Bildung der Modulnote:

Klausur und Bericht bis zu 25%

Modulbeschreibung Wahlpflichtmodul 1 und 2

Schlüsselworte: Vertiefung im eigenen Studienprofil

Zielgruppe:	1. Semester AIM	Modulnummer:	AI 103 AI 104
Arbeitsaufwand:	5 ECTS	Abhängig vom gewählten Modul	
Davon	Kontaktzeit	Abhängig vom gewählten Modul	
	Selbststudium	Abhängig vom gewählten Modul	
	Prüfungsvorbereitung	Abhängig vom gewählten Modul	
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Dr. Andreas Rößler		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Abhängig vom gewählten Modul

Gesamtziel:

Die Studierenden erlangen eine wissenschaftliche und fachliche Vertiefung auf dem Gebiet der Vertiefung.

Inhalt:

Abhängig vom gewählten Modul

Literaturhinweise:

Abhängig vom gewählten Modul

Wird angeboten:

Die zur Auswahl stehenden Wahlfächer werden zu Semesterbeginn öffentlich bekannt gegeben.

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Abhängig vom gewählten Modul
Leistungskontrolle:	Abhängig vom gewählten Modul
Anteil Semesterwochenstunden:	Abhängig vom gewählten Modul
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernergebnisse:

Abhängig vom gewählten Modul

Bildung der Modulnote:

Abhängig vom gewählten Modul

Modulbeschreibung Forschungsprojekt 1

Schlüsselworte: Wissenschaftliches Arbeiten im Team

Zielgruppe:	1. Semester AIM	Modulnummer:	AI 105
Arbeitsaufwand:	10 ECTS		300 h
Davon	Kontaktzeit		30 h
	Selbststudium		270 h
	Prüfungsvorbereitung		0 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Dr. Andreas Rößler		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Anwendung der Methoden des Softwareentwicklung, Kenntnisse in der gewählten Vertiefungsrichtung, Grundkenntnisse wissenschaftlichen Arbeitens

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Qualitätskriterien für wissenschaftliches Arbeiten
- die Methoden des Projektmanagements

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- wissenschaftliche Projekte im Team zu planen und durchzuführen,
- die in den Kern- und Vertiefungsfächern erworbenen Kenntnisse zur Lösung von Problemen aus dem Bereich der Forschung einzusetzen,
- Lösungsansätze (Stand der Technik) zu recherchieren und zu verstehen,
- gefundene Lösungsansätze bewerten,
- die Ergebnisse ihres wissenschaftlichen Arbeitens nachvollziehbar dokumentieren.

Übergreifende Kompetenzen

Die Studierenden können

- unter Anleitung komplexe Problemstellungen aus der Forschung oder aus der Industrie innerhalb einer vorgegebenen Frist zu lösen,
- neue Kenntnisse in der Informatik zu gewinnen und neue Verfahren zu entwickeln,
- Wissen aus verschiedenen Domänen integrieren,
- in einem Team gemeinsam eine Aufgabe erfolgreich umzusetzen.

Inhalt:

Im Forschungsprojekt bearbeiten Studierende in einem Team unter Anleitung eines Dozenten aktuelle Forschungsthemen aus wissenschaftlichen Einrichtungen oder forschungsnahe Themen aus der Industrie. Die Projekte sind auf ein Jahr angelegt, wobei alle Phasen eines Softwareprojekts durchlaufen werden sollen: Problem- und Anforderungsanalyse, Recherche des Standes der Technik, Projektplanung, Erarbeitung von Lösungsansätzen, Softwareentwurf, Implementierung, Testphase. Die Studierenden erarbeiten Arbeits- und Zeitpläne und berichten regelmäßig über ihren Fortschritt. Am Ende der Semester tragen die Studierenden jeweils Zwischen- bzw. Endergebnisse vor.

Literaturhinweise:

Abhängig von der gewählten Problemstellung

Wird angeboten:

Sommer- und Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Projektarbeit mit Bericht und Referat
Leistungskontrolle:	Schriftlicher Bericht und Referat (20 Minuten)
Anteil Semesterwochenstunden:	2 SWS
Geschätzte studentische Arbeitszeit:	300 Stunden

Bildung der Modulnote:

Projektarbeit: Referat und Dokumentation

Modulbeschreibung Advanced Software Engineering

Schlüsselworte: Software Engineering, Metriken, Empirie

Zielgruppe:	2. Semester AIM	Modulnummer:	AI 201
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Englisch oder Deutsch		
Modulverantwortung:	Dipl.-Inf. Holger Cermann		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Grundlagen des Softwareengineering

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- Agile Vorgehensweisen in Großprojekten: Scrum/Agile@Scale, Agile Requirements Engineering
- Architekturmodelle für große Systeme: (a)synchrone Kommunikation, Batches, Transaktionen
- Aktuelle Methoden der Software-Qualität, Software Measurements und Metriken, Testing
- Verfahren der Software-Archäologie
- Securityaspekte bei der Systementwicklung

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Methoden des Projekt- & Risikomanagement in realen Softwareprojekten anzuwenden
- Unternehmensarchitekturen zu entwickeln: SOA, Microservices, Governance, Software in der Cloud, DevOps
- an der Ausgestaltung und Umsetzung von Architektur und SE-Prozessen von realen Projekten mitzuwirken

Übergreifende Kompetenzen

Die Studierenden können

- Probleme des Softwareengineerings in realen modernen Projekten lösen
- Arbeiten in global verteilten Teams durchführen

Inhalt:

Fokus auf wichtige aktuelle Aspekte mit Praxisbezug zu Architektur, Qualität, Vorgehen und Digitale Transformation im Software Engineering

Literaturhinweise:

Somerville: Software Engineering. Pearson 2012.
Ludewig, Lichter: Software Engineering. dpunkt.verlag 2013.

Wird angeboten:

in jedem Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform: Vorlesung mit Übungen

Leistungskontrolle: Klausur (90 Minuten)

Anteil Semesterwochenstunden: 4 SWS

Geschätzte studentische Arbeitszeit: 150 Stunden

Lernergebnisse:

Die Studierenden sind in der Lage das theoretisch erlangte Wissen aus der Vorlesung in Übungen inkl. Hackathon anzuwenden und auch vor der Gruppe vorzustellen.

Bildung der Modulnote:

Klausur

Modulbeschreibung IT Management

Schlüsselworte: IT Management

Zielgruppe:	2. Semester AIM	Modulnummer:	AI 202
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr. Thomas Rodach		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Grundkenntnisse in Betriebswirtschaft sowie von Informationssystemen in Unternehmen

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Bedeutung der IT-Strategie
- Methoden, die IT-Strategie aus der Geschäftsstrategie zu entwickeln und deren Wertbeitrag einzuschätzen
- unterschiedliche Organisationsformen einer IT-Abteilung
- Aufgaben und Methoden des IT-Controlling
- die Bedeutung von IT-Governance und – Compliance

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- die Auswirkungen unterschiedlicher Organisationsformen einer IT-Abteilung einzuschätzen
- IT-Controlling Konzepte zu entwickeln
- Methoden und Vorgehensweisen zur Entwicklung und Management einer IT Unternehmensarchitektur einzusetzen

Übergreifende Kompetenzen

Die Studierenden können

- eine IT-Unternehmensarchitektur entwickeln und das sich daraus ergebende Life Cycle Management der Anwendungssysteme konzipieren
- IT-Governance und – Compliance Konzepte entwickeln
- Managementaufgaben in der IT, die im Rahmen von Führungsfunktionen wahrzunehmen sind, übernehmen

Inhalt:

- IT-Strategie und IT Business Alignment
- Aufbau und Organisation einer IT-Abteilung
- IT-Controlling
- IT-Unternehmensarchitektur und IT-Life Cycle Management
- IT-Governance und -Compliance

Literaturhinweise:

Gadatsch, Mayer: Masterkurs IT-Controlling, 5. Auflage, 2014
Hofmann, Schmidt: Masterkurs IT-Management, 2. Auflage, 2010
Krcmar: Informationsmanagement, 5. Auflage, 2010

Wird angeboten:

in jedem Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen
Leistungskontrolle:	Mündliche Prüfung (20 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernergebnisse:

Die Studierenden

- verstehen die Bedeutung der IT-Strategie und kennen Methoden, die IT-Strategie aus der Geschäftsstrategie zu entwickeln und deren Wertbeitrag einzuschätzen
- sind mit unterschiedlichen Organisationsformen einer IT-Abteilung vertraut und können deren Auswirkungen einschätzen
- kennen Aufgaben und Methoden des IT-Controlling und können ein IT-Controlling Konzept entwickeln
- können eine IT-Unternehmensarchitektur entwickeln und das sich daraus ergebende Life Cycle Management der Anwendungssysteme konzipieren
- verstehen die Bedeutung von IT-Governance und – Compliance und können entsprechende Konzepte entwickeln
- können eine komplexe Problemstellung systematisch analysieren
- sind in der Lage, die Methoden entsprechend dem aktuellen Stand der Wissenschaft darauf anzuwenden
- verstehen es, Recherchemethoden nach neuen Forschungsergebnissen einzusetzen
- sind in der Lage, die Anwendbarkeit der Methoden in der Praxis kritisch einzuschätzen, Methoden weiterzuentwickeln oder eigene Methoden zu entwickeln
- können daraus eine eigenständige Problemlösung entwickeln

Bildung der Modulnote:

Mündliche Prüfung (20 Minuten)

Modulbeschreibung Wahlpflichtmodul 3 und 4

Schlüsselworte: Vertiefung im eigenen Studienprofil

Zielgruppe:	1. Semester AIM	Modulnummer:	AI 203 AI 204
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit	Abhängig vom gewählten Modul	
	Selbststudium	Abhängig vom gewählten Modul	
	Prüfungsvorbereitung	Abhängig vom gewählten Modul	
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Dr. Andreas Rößler		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Abhängig vom gewählten Modul

Gesamtziel:

Die Studierenden erlangen eine wissenschaftliche und fachliche Vertiefung auf dem Gebiet der Vertiefung.

Inhalt:

Abhängig vom gewählten Modul

Literaturhinweise:

Abhängig vom gewählten Modul

Wird angeboten:

Die zur Auswahl stehenden Wahlfächer werden zu Semesterbeginn öffentlich bekannt gegeben.

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Abhängig vom gewählten Modul
Leistungskontrolle:	Abhängig vom gewählten Modul
Anteil Semesterwochenstunden:	Abhängig vom gewählten Modul
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernergebnisse:

Abhängig vom gewählten Modul

Bildung der Modulnote:

Abhängig vom gewählten Modul

Modulbeschreibung Forschungsprojekt 2

Schlüsselworte: Wissenschaftliches Arbeiten im Team

Zielgruppe:	2. Semester AIM	Modulnummer:	AI 205
Arbeitsaufwand:	10 ECTS		300 h
Davon	Kontaktzeit		30 h
	Selbststudium		270 h
	Prüfungsvorbereitung		0 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Dr. Andreas Rößler		
Stand:	01.03.2020		

Voraussetzungen nach Prüfungsordnung:

Keine

Empfohlene Voraussetzungen:

Anwendung der Methoden des Softwareentwicklung, Kenntnisse in der gewählten Vertiefungsrichtung, Grundkenntnisse wissenschaftlichen Arbeitens

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Qualitätskriterien für wissenschaftliches Arbeiten
- die Methoden des Projektmanagement

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- wissenschaftliche Projekte im Team zu planen und durchzuführen,
- die in den Kern- und Vertiefungsfächern erworbenen Kenntnisse zur Lösung von Problemen aus dem Bereich der Forschung einzusetzen,
- Lösungsansätze (Stand der Technik) zu recherchieren und zu verstehen,
- gefundene Lösungsansätze bewerten,
- die Ergebnisse ihres wissenschaftlichen Arbeitens nachvollziehbar dokumentieren.

Übergreifende Kompetenzen

Die Studierenden können

- unter Anleitung komplexe Problemstellungen aus der Forschung oder aus der Industrie innerhalb einer vorgegebenen Frist zu lösen,
- neue Kenntnisse in der Informatik zu gewinnen und neue Verfahren zu entwickeln,
- Wissen aus verschiedenen Domänen integrieren,
- in einem Team gemeinsam eine Aufgabe erfolgreich umzusetzen.

Inhalt:

Im Forschungsprojekt bearbeiten Studierende in einem Team unter Anleitung eines Dozenten aktuelle Forschungsthemen aus wissenschaftlichen Einrichtungen oder forschungsnahe Themen aus der Industrie. Die Projekte sind auf ein Jahr angelegt, wobei alle Phasen eines Softwareprojekts durchlaufen werden sollen: Problem- und Anforderungsanalyse, Recherche des Standes der Technik, Projektplanung, Erarbeitung von Lösungsansätzen, Softwareentwurf, Implementierung, Testphase. Die Studierenden erarbeiten Arbeits- und Zeitpläne und berichten regelmäßig über ihren Fortschritt. Am Ende der Semester tragen die Studierenden jeweils Zwischen- bzw. Endergebnisse vor.

Literaturhinweise:

Abhängig von der gewählten Problemstellung

Wird angeboten:

Sommer- und Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Projektarbeit mit Bericht und Referat
Leistungskontrolle:	Schriftlicher Bericht und Referat (20 Minuten)
Anteil Semesterwochenstunden:	2 SWS
Geschätzte studentische Arbeitszeit:	300 Stunden

Bildung der Modulnote:

Projekt: Referat und Dokumentation

Modulbeschreibung Masterarbeit

Schlüsselworte: Selbstständiges wissenschaftliches Arbeiten

Zielgruppe: 3. Semester AIM

Modulnummer: AI 301

Arbeitsaufwand:	25 ECTS	750 h
Davon	Kontaktzeit	30 h
	Selbststudium	660 h
	Prüfungsvorbereitung	60 h
Unterrichtssprache:	Deutsch oder Englisch	
Modulverantwortung:	Prof. Dr. Andreas Rößler	

Stand: 01.03.2020

Empfohlene Voraussetzungen:

Kenntnisse der Methoden wissenschaftlichen Arbeitens, sichere Anwendung der Methoden des Softwareengineering, umfassende Kenntnisse in der gewählten Vertiefungsrichtung

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Qualitätskriterien für wissenschaftliches Arbeiten
- die Methoden des Projektmanagement

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- wissenschaftliche Fragestellungen zu formulieren,
- wissenschaftliche Methoden anzuwenden,
- wissenschaftliche Projekte zu planen und durchzuführen,
- die in den Kern- und Vertiefungsfächern erworbenen Kenntnisse zur Lösung von Problemen einzusetzen,
- Lösungsansätze (Stand der Forschung) zu recherchieren und zu verstehen, und zu bewerten,
- eigene Lösungsansätze zu entwickeln und umzusetzen,
- die Ergebnisse ihres wissenschaftlichen Arbeitens nachvollziehbar dokumentieren.

Übergreifende Kompetenzen

Die Studierenden können

- eine komplexe Problemstellung der Informatik selbstständig, wissenschaftlich, innerhalb einer vorgegebenen Frist zu bearbeiten,
- den dazugehörigen Stand der Forschung zu recherchieren, zu strukturieren und zu verstehen,
- geeignete Methoden und Verfahren auszuwählen, diese korrekt einzusetzen und falls notwendig sie anzupassen oder weiter zu entwickeln,
- ihre Ergebnisse mit anderen Ergebnissen zu vergleichen und ihre Lösungsansätze kritische zu überprüfen,
- ihre Ergebnisse strukturiert zu dokumentieren und in wissenschaftlicher Form zu veröffentlichen.

Inhalt:

- Problemanalyse und Eingrenzung des Themas
- Literaturrecherche
- Planung der Vorgehensweise, Erarbeitung eines Lösungsansatzes
- Zeit- und Projektmanagement
- Herstellen eines Bezugs zwischen eigenen Ansätze und dem Stand der Forschung
- Wissenschaftliche Darstellung der Ergebnisse
- Verteidigung

Literaturhinweise:

Abhängig von der gewählten Problemstellung

Wird angeboten:

in jedem Semester

Teilgebiete und Leistungsnachweise:

Modulteil:	Masterarbeit
Lehr- und Lernform:	Wissenschaftliche Arbeit mit Bericht und Verteidigung
Leistungskontrolle:	Schriftlicher Bericht und Referat (20 Minuten)
Anteil Semesterwochenstunden:	2 SWS
Geschätzte studentische Arbeitszeit:	750 Stunden
Lernergebnisse:	

Die Studierenden verfügen über **Kenntnisse** der grundlegenden Techniken des wissenschaftlichen Arbeitens, insbesondere Gliederung, Zitieren und formale Aspekte. Sie besitzen die **Fertigkeiten** Probleme zu analysieren, zu strukturieren und zu lösen. Dazu sind sie in der Lage, den dazugehörigen Stand der Forschung zu recherchieren, zu strukturieren und zu verstehen sowie geeignete Methoden und Verfahren auszuwählen, diese korrekt einzusetzen und falls notwendig sie anzupassen oder weiter zu entwickeln. Sie können ihre Ergebnisse mit anderen Ergebnissen vergleichen und ihre Lösungsansätze kritisch überprüfen. Die Studierenden verfügen über die **Kompetenzen** zur selbstständigen Projekt- und Zeitplanung und haben Verantwortungsbewusstsein in Bezug auf wissenschaftliche Qualitätskriterien wie Überprüfbarkeit und Verlässlichkeit von wissenschaftlichen Ergebnissen erworben.

Bildung der Modulnote:

Gemittelte Note aus Bericht (Faktor 22) und Referat (Faktor 3)

Modulbeschreibung Publikation

Schlüsselworte: Selbstständiges wissenschaftliches Schreiben

Zielgruppe:	3. Semester AIM	Modulnummer:	AI 302
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		15 h
	Selbststudium		135 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Dr. Andreas Rößler		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Kenntnisse der Methoden wissenschaftlichen Arbeitens, erfolgreiche Teilnahme am Forschungsprojekt (AI 105 und AI 205)

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die formalen Aspekte einer wissenschaftlichen Veröffentlichung
- geeignete Journalen und Konferenzen

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- eine Thema für eine Veröffentlichung einzugrenzen
- den Stand der Forschung zu recherchieren, zu strukturieren, zu verstehen und wiederzugeben
- Bezüge zwischen eigenen Ansätzen und dem Stand der Forschung herzustellen

Übergreifende Kompetenzen

Die Studierenden können

- Forschungsergebnisse strukturiert dokumentieren und in eine publikationsreife Form bringen.

Literaturhinweise:

Balzert, Helmut; Schröder, Marion; Schaefer, Christian (2013): Wissenschaftliches Arbeiten. Ethik Inhalt & Form wiss. Arbeiten Handwerkszeug Quellen Projektmanagement Präsentation. 2. Aufl., 1. korr. Nachdr. Herdecke, Witten: W3L-Verl. (Soft skills).

Kornmeier, Martin (2013): Wissenschaftlich schreiben leicht gemacht. Für Bachelor Master und Dissertation. 6., aktualisierte Aufl. Bern, [Stuttgart]: Haupt (UTB, 3154 : Schlüsselkompetenzen).

Theisen, Manuel René (2013): Wissenschaftliches Arbeiten. Erfolgreich bei Bachelor- und Masterarbeit. 16., vollst. überarb. Aufl. München: Vahlen.

Wird angeboten:

in jedem Semester

Lehr- und Lernform:	Verfassen einer wissenschaftlichen Publikation
Leistungskontrolle:	Veröffentlichungsreifer Text
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernergebnisse:

Die Studierenden verfügen über Kenntnisse darüber, wie und wo wissenschaftliche Ergebnisse publiziert werden können. Sie sind in der Lage, die Ergebnisse ihrer wissenschaftlichen Tätigkeit nachvollziehbar nach wissenschaftlichen Kriterien zusammenzufassen. Sie verfügen über die Kompetenz, selbstständig für die Qualitätssicherung der Publikation zu sorgen und diese fristgerecht einzureichen.

Bildung der Modulnote:

Bewertung einer veröffentlichungsreifen Publikation

Modulbeschreibung Intelligent Data Analytics

Schlüsselworte: Big Data, Data Mining, Zeitreihen, Klassifikation, Vorhersage, Querying

Zielgruppe:	1. oder 2. Semester AIM	Modulnummer:	AI W01
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Englisch		
Modulverantwortung:	Prof. Dr. Gabriele Gühring		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Gute mathematische Kenntnisse v.a. in Statistik und Optimierung
Kenntnisse in Informatik

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- Grundlagen der Zeitreihen
- Anwendungen, in denen Zeitreihen generiert und aufgezeichnet werden
- Verfahren der Klassifikation von Zeitreihendaten
- Verfahren zur Regressionsanalyse und zur Vorhersage
- Grundlagen der künstlichen Neuronalen Netze

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- geeignete Analyseverfahren auszuwählen und anzuwenden

Übergreifende Kompetenzen

Die Studierenden können

- Zeitreihen mit Hilfe von Algorithmen aus den Bereichen „Data Mining“ und „maschinelles Lernen“ intelligent analysieren.

Inhalt:

- Introduction to Data Mining with a focus on Time Series Data (Temporal Data Mining)
- Fundamentals of Time Series Data
- Classification, Time Series Querying, Regression/Forecasting
- Visualization of Time Series
- Artificial Neural Networks
- Applied Data Mining for Hybrid Vehicle Powertrain

Literaturhinweise:

T. Mitsa: Temporal Data Mining. Chapman & Hall/CRC Data Mining and Knowledge Discovery. 2010

J. Han, M. Kamber, J. Pei: Data Mining – Concepts and Techniques (3rd Edition). Morgan Kaufman, 2012

R. J. Hyndman, G. Athanasopoulos: Forecasting: principles and practice. Available online at <https://www.otexts.org/fpp>, 2014

Wird angeboten:

Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit praktischen Übungen
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernergebnisse:

Die Studierenden besitzen Grundkenntnisse in „Data Mining auf Zeitreihen“ und im Umgang mit der Software „R“. Sie sind in der Lage, ausgewählte Verfahren aus den Funktionalitäten „Querying“, „Klassifikation“ und „Vorhersage“ auf Zeitreihen anzuwenden. Diese Techniken werden in vielen industriellen Anwendungen, z.B. bei einer Modellbasierten Diagnose der Hochvoltbatterie eines Hybridfahrzeuges, eingesetzt. Die gelernten Methoden und Konzepte können zum Zwecke des „Data Mining“ auch auf andere Datentypen angewandt werden.

Bildung der Modulnote:

Klausur

Modulbeschreibung Business Intelligence

Schlüsselworte: Business Intelligence, Datawarehouse, OLAP, Business Performance

Zielgruppe:	1. oder 2. Semester AIM	Modulnummer:	AI W02
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		30 h
	Selbststudium		90 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr. Dirk Hesse		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Grundlagen Datenbanken

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Grundbegriffe der Business Intelligence
- die Bedeutung von BI für die unternehmerische Praxis
- verschiedene Konzepte und Methoden der BI

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- verschiedene Instrumente und Anwendungen der Business Intelligence zielgerichtet einzusetzen

Übergreifende Kompetenzen

Die Studierenden können

- für spezifische Anwendungsfelder geeignete BI-Verfahren auswählen und umsetzen.

Inhalt:

Vertiefung der Methodenkompetenz im Bereich Business Analytics:

- Verarbeitung (semi-) strukturierter Daten im ETL Prozess
- Logische Modellierung (Star Schema, Snowflake Schema etc.)
- Einrichtung von multidimensionalen Modellen (OLAP Cubes)
- Reporting und Analyse mittels verschiedener Tools, Queries und Webreports
- Performanceverbesserungen und Berechtigungskonzepte

Literaturhinweise:

Business Intelligence and Analytics: Systems for Decision Support, Sharda, Turba, Delen, Pearson Education Limited, 10. Auflage 2014.

Kemper, Hans-Georg, et al.: Business Intelligence - Grundlagen und praktische Anwendungen. Eine Einführung in die IT-basierte Managementunterstützung, Vieweg und Teubner, 3. Auflage 2010.

Wird angeboten:

Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Seminar
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	1,5 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Lernergebnisse:

Die Studierenden verfügen über Kenntnisse der grundlegenden Konzepte des Business Intelligence. Sie haben die Fertigkeit verschiedene Ansätze, Methoden und Werkzeuge des Business Intelligence zu unterscheiden und können Unternehmens- /Wettbewerbs- und Kundendaten analysieren. Sie verfügen über die Kompetenz, die vorgestellten Konzepte in das unternehmensweite Informations- und Wissensmanagement zu integrieren.

Lehr- und Lernform:	Projektarbeit (Labor)
Leistungskontrolle:	Referat, Testat
Anteil Semesterwochenstunden:	0,5 SWS
Geschätzte studentische Arbeitszeit:	60 Stunden

Lernergebnisse:

Die Studierenden haben die Fertigkeit BI-Systeme zu modellieren und zu implementieren. Insbesondere beherrschen sie verschiedene BI-Konzeptionen und können diese im Rahmen betriebswirtschaftlicher Fragestellungen anwenden und in BI-Planungssysteme eigenständig umsetzen.

Bildung der Modulnote:

Klausur

Modulbeschreibung Mobile Communication

Schlüsselworte: Automotive, Communication, Safety, Security

Zielgruppe:	1. oder 2. Semester AIM	Modulnummer:	AI W03
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Englisch		
Modulverantwortung:	Prof. Dr.-Ing. Harald Melcher, Prof. Dr. Dominik Schoop		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Grundlagen Kommunikationstechnik
Module IT-Security Engineering and Advanced Software Engineering

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Grundlagen der Intelligent Transportation Systems (ITS) und der Vehiculäre Ad-Hoc-Netzwerke (VANETs)
- C2I/V2I-Anwendungen
- Automotive wireless Netztechnologie (WLAN (IEEE 802.11p), CAM, DENM)
- relevante Standards (IEEE, ISO)
- Systeme zur Positionsbestimmung (GPS, ...)
- den Zusammenhang Safety und Security

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- die Ziele von ITS und VANETs zu erklären
- die Sicherheit von VANETs einzuschätzen und Sicherheitsmaßnahmen vorzuschlagen
- die Architektur und Technologie von VANETs zu erklären

Übergreifende Kompetenzen

Die Studierenden können

- einfache Car2Infrastructure- und Car2-Car-Applikationen zu implementieren.

Inhalt:

Vertiefung der Methodenkompetenz im Bereich Business Analytics:

- Verarbeitung (semi-) strukturierter Daten im ETL Prozess
- Logische Modellierung (Star Schema, Snowflake Schema etc.)
- Einrichtung von multidimensionalen Modellen (OLAP Cubes)
- Reporting und Analyse mittels verschiedener Tools, Queries und Webreports
- Performanceverbesserungen und Berechtigungskonzepte

Literaturhinweise:

Christoph Sommer, Falko Dressler: Vehicular Networking. Cambridge University Press, 2014

Erdal Cayirci, Chunming Rong: Security in Wireless Ad Hoc and Sensor Networks, John Wiley & Sons, 2009

Srikanta Patnaik, Xiaolong Li, Yeon-Mo Yang: Recent Development in Wireless Sensor and Ad-hoc Networks, Springer, 2014

Wird angeboten:

Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Seminar und Projektarbeit
Leistungskontrolle:	Benotetes Referat
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	60 Stunden

Lehr- und Lernform:	Projektarbeit (Labor)
Leistungskontrolle:	Unbenotetes Referat, Testat
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Bildung der Modulnote:

Projektarbeit mit Referat (20 Minuten)

Modulbeschreibung Datenfusion

Schlüsselworte: Datafusion, Kalman-Filter, Bayes Netze, Markov Modelle

Zielgruppe:	1. oder 2. Semester AIM	Modulnummer:	AI W04
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		45 h
	Selbststudium		75 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Reiner Marchthaler		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Grundlagen Statistik und Automatentheorie (im Speziellen in Zustandsautomaten)

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Grundlagen der Kalman-Filter-Gleichungen
- die hier zugrundeliegenden Annahmen
- die Grundlagen von Markov-Modellen, Partikelfilter und Bayes-Netzen

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage Zustandsmodelle zu modellieren

Übergreifende Kompetenzen

Die Studierenden können in praktischen Übungen Kalman-Filter in einem Mikrocontroller implementieren

Inhalt:

- Kalman-Filter und Partikelfilter
- Markov Decision Processes und Hidden Markov Modelle
- Bayes Netzwerke

Literaturhinweise:

Richard E. Neapolitan: "Learning Bayesian Networks". Prentice Hall, 2003

Olle Häggström: "Finite Markov Chains and Algorithmic Applications". Cambridge University Press, 2002

Brown, R. G.; Hwang, P. Y. C.: "Introduction to Random Signals and Applied Kalman Filtering". Third Edition. John Wiley & Sons, Inc., 2012

Simon, Dan "Optimal State Estimation", John Wiley & Sons, Inc., 2006

Thrun, S.; u.a. "Probabilistic Robotics", MIT Press, 2005

Marchthaler, R.; Dingler S. "Kalman-Filter - Einführung in die Zustandsschätzung und ihre Anwendung", Springer-Vieweg Verlag, 2016

Wird angeboten:

Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernergebnisse:

Die Studierenden beherrschen die mathematischen Konzepte zum Entwurf eines Kalman-Filters, einer Markov-Kette und eines Bayes-Netzwerkes. Darüber hinaus sind Sie in der Lage selbstständig ein Kalman-Filter für zur Sensordatenfusion zu entwerfen.

Bildung der Modulnote:

Klausur

Modulbeschreibung Automotive System Design

Schlüsselworte: System- und Softwarearchitekturen sowie Netzwerke und Kommunikationsprotokolle in der automobilen Anwendung, Funktionale Sicherheit

Zielgruppe:	1. oder 2. Semester AIM	Modulnummer:	AI W05
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Englisch oder Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Werner Zimmermann		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Kenntnisse der Informatik und Netzwerke auf Bachelorniveau, Kompetenzen zur Softwareentwicklung in einer gängigen Programmiersprache bevorzugt, C/C++

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Grundlagen der Anwendungsdomänen (Powertrain, Chassis, Body, Infotainment, Driver Assistance)
- Grundlagen verteilter Systeme, Kommunikationsanforderungen, Analyse und Gestaltung unter Echtzeit- und Sicherheitsanforderungen
- die Architektur von Systemen im Automobil und deren Software

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Analysen der Zuverlässigkeit und der Sicherheit durchzuführen
- funktionale Sicherheitsanforderungen aufzustellen zu implementieren
- die Netzwerkkommunikation von Systemen im Automobil zu gestalten und quantitativ abzuschätzen

Übergreifende Kompetenzen

Die Studierenden können

- Systeme im Automobil hinsichtlich Sicherheit und Zuverlässigkeit abschätzen
- sichere und zuverlässige Systeme im Automobil konzipieren und implementieren

Literaturhinweise:

J. Schäufele, T. Zurawka: Automotive Software Engineering, Springer-Vieweg
C. Marscholik, P. Supke: Road Vehicles – Diagnostic Communication. VDE-Verlag
W. Zimmermann, R. Schmidgall: Bussysteme in der Fahrzeugtechnik – Protokolle, Standards und Softwarearchitektur. Springer-Vieweg

Wird angeboten:

Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung
Leistungskontrolle:	Klausur (60 Minuten)
Anteil Semesterwochenstunden:	3 SWS
Geschätzte studentische Arbeitszeit:	120 Stunden

Lernergebnisse:

Die Studierenden kennen

- die Grundlagen der Anwendungsdomänen (Powertrain, Chassis, Body, Infotainment, Driver Assistance)
- Grundlagen verteilter Systeme, Kommunikationsanforderungen, Analyse und Gestaltung unter Echtzeit- und Sicherheitsanforderungen
- die Architektur von Systemen im Automobil und deren Software

Lehr- und Lernform:	Gruppenprojekt und Referat
Leistungskontrolle:	Referat (20 Minuten)
Anteil Semesterwochenstunden:	1 SWS
Geschätzte studentische Arbeitszeit:	30 Stunden

Lernergebnisse:

Die Studierenden sind in der Lage

- Analysen der Zuverlässigkeit und der Sicherheit durchzuführen
- funktionale Sicherheitsanforderungen aufzustellen zu implementieren
- die Netzwerkkommunikation von Systemen im Automobil zu gestalten und quantitativ abzuschätzen

Die Studierenden können

- Systeme im Automobil hinsichtlich Sicherheit und Zuverlässigkeit abschätzen
- sichere und zuverlässige Systeme im Automobil konzipieren und implementieren

Bildung der Modulnote:

Klausur

Modulbeschreibung Advanced Control

Schlüsselworte: Fuzzy-Regelung; moderne Regelungstechnik (PI-Zustandsregler, Zustands- und Störgrößenbeobachter), Optimale Regler und Zustandsschätzer

Zielgruppe:	1. oder 2. Semester AIM	Modulnummer:	AI W06
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr. Walter Lindermeir		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Kenntnisse der Booleschen Schaltalgebra, der Wahrscheinlichkeitsrechnung sowie der Matrix-Vektor-Rechnung, Integral- und Differentialrechnung.
Kenntnisse der Regelungstechnik (PID-Regler, stationäres Verhalten von Regelkreisen, Blockschaltbildarithmetik, Zustandsregler und Luenberger Beobachter)
Beschreibung dynamischer Systeme im Frequenz (Laplace-Übertragungsfunktion) und Zeitbereich (Zustandsbeschreibung)
Kenntnisse einer Simulationssprache (z.B.. MATLAB/Simulink)

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Grundlagen der Fuzzy-Logik (Einsatzgebiete, Fuzzifizierung, Regelbasis, Defuzzifizierung)
- Einsatz und Funktionsweise von Fuzzy-Reglern
- Entwurf und Einsatz von PI-Zustandsreglern und Störgrößenbeobachtern (Polvorgabe)
- Optimale Regler (LQR-Regelung) und optimale Zustandsschätzung dynamischer Systeme (Kalman-Filter)
- Regler in der Zwei-Freiheitsgrad-Struktur (2-DOF)
- Steuerungen bzw. Vorsteuerungen basierend auf der Flachheitsmethode
- IMC-Regler

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Fuzzy-Inferenz-Systeme zu entwerfen und einzusetzen, z.B. unter Einsatz der Matlab Fuzzy-Logic-Toolbox
- PI-Zustandsregler zu entwickeln (einfache Systeme von Hand, bzw. unter Einsatz von Matlab/Simulink) und in Betrieb zu nehmen
- Störgrößenmodelle zu definieren und Störgrößenbeobachter zu entwickeln
- Kalman-Filter zu entwerfen und sinnvoll zu parametrieren
- 2-DOF-Reglerstrukturen zu verstehen und z.B. mit Hilfe der Flachheitsmethodik auslegen
- IMC-Regler zu entwerfen

Übergreifende Kompetenzen

Die Studierenden können

- die in der Systemtheorie gebräuchlichen Systembeschreibungen (Laplace-Übertragungsfunktionen und Zustandsbeschreibungen) vorteilhaft einsetzen
- Regelungen (PID-/Zustands-/IMC-/Fuzzy-Regler) eigenständig auslegen
- Zustandsschätzer (Luenberger, Kalman, Störgrößen) eigenständig auslegen

- Erweiterte Regler-Strukturen (2-DOF-Struktur, IMC) sinnvoll einsetzen
- Sich im Umfeld von eingebetteten Systemen z.B. in den Bereichen der Automobiltechnik oder der Industrieautomatisierung fachlich einbringen

Literaturhinweise:

Föllinger, O.: Regelungstechnik. Hüthig Verlag
 Lunze, J.: Regelungstechnik. 2 Bände, Springer Verlag
 Unbehauen, H.: Regelungstechnik. 3 Bände, Vieweg Verlag
 Simon, D.: Optimal State Estimation
 Mann, Schiffelgen, Froriep: Einführung in die Regelungstechnik

Wird angeboten:

Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform: Vorlesung
Leistungskontrolle: Klausur (90 Minuten)
Anteil Semesterwochenstunden: 4 SWS
Geschätzte studentische Arbeitszeit: 120 Stunden

Lernergebnisse:

Die Studierenden beherrschen die Modellierung von dynamischen Systemen in der Zustandsdarstellung und können die in der Systemtheorie gebräuchlichen Systembeschreibungen (Laplace-Übertragungsfunktionen und Zustandsbeschreibungen) vorteilhaft einsetzen. Die Studierenden können PID, P- und PI-Zustandsregler sowie Fuzzy-Regler und erweiterte Regler-Strukturen (2-DOF, IMC) auslegen und deren jeweilige Vor- und Nachteile für eine Anwendung einschätzen. Die Studierenden sind in der Lage, Zustandsschätzer (Luenberger, Kalman, Störgrößen) zu entwerfen und in Projekten einzusetzen. Dies gilt jeweils sowohl für die analytische als auch für die numerische (Matlab/Simulink) Auslegung der jeweiligen Systeme.

Bildung der Modulnote:

Klausur

Modulbeschreibung Web Services

Schlüsselworte: Serviceorientierte Architekturen, Web-Anwendungen

Zielgruppe:	1. oder 2. Semester AIM	Modulnummer:	AI W12
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		90 h
	Prüfungsvorbereitung		0 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Dr.-Ing. Andreas Rößler		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Grundlegende Web-Technologien, Rechnernetze, Softwareentwicklung

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- Grundlagen von Web Services
- Methoden zur Identifikation und Spezifikation von Web Services
- Konzepte von HTTP, SOAP und WSDL
- Konzepte von REST
- Best Practices für den Entwurf von REST Web Services
- Methoden zur Absicherung von Web Services
- Varianten für den Betrieb von Web Services (bspw. Docker Images)
- Methoden zur Entwicklung von REST Web Service Clients
- Grundlagen serviceorientierter Architekturen
- Unterschiede zwischen serviceorientierten Architekturen als Integrations- und strategische IT-Architekturen
- Grundlagen des Business Process Management

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Web Services auf Basis von SOAP zu entwickeln
- REST Web Services unter Berücksichtigung aktueller Best Practices zu entwickeln
- REST Web Services als Microservices in Docker Images zu deployen
- Web Service Clients zu entwickeln

Übergreifende Kompetenzen

Die Studierenden können

- verteilte Web-Architekturen mit Hilfe von Web Services konzipieren

Inhalt:

- Grundlagen der Web Services
- Spezifikation von Web Service Requirements
- Web Services mit SOAP und WSDL
- Web Services mit REST
- Best Practices für REST Web Services
- Web Service Security
- Web Service Deployment
- Entwicklung von Web Service Clients
- Serviceorientierte Architekturen und Business Process Management

Literaturhinweise:

Erl, Th. et.al.: SOA with REST. Prentice Hall 2012.

Erl, Th.: Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall 2015

Papazoglou, M.P.: Web Services: Principles and Technology, Pearson Education, 2008.

Wird angeboten:

Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung / Seminar
Leistungskontrolle:	Referat (20 Minuten)
Anteil Semesterwochenstunden:	2 SWS
Geschätzte studentische Arbeitszeit:	90 Stunden

Lernergebnisse:

Die Studierenden kennen die wichtigsten Technologien und Standards zur Entwicklung von Web Services. Sie sind in der Lage die Architektur serviceorientierter Web-Anwendungen zu verstehen, zu beurteilen und anzuwenden.

Lehr- und Lernform:	Projektarbeit
Leistungskontrolle:	Referat (20 Minuten)
Anteil Semesterwochenstunden:	2 SWS
Geschätzte studentische Arbeitszeit:	60 Stunden

Lernergebnisse:

Die Studierenden können Anforderungen an verteilte Web-Architekturen und Web Services modellieren und die Technologien und Werkzeuge für Web Services in Projekten anwenden.

Bildung der Modulnote:

Referat (3) und Projektarbeit (2)

Modulbeschreibung Mobile Applikationen

Schlüsselworte: Apps, Client, Server, Soziale Netze

Zielgruppe:	1. oder 2. Semester AIM	Modulnummer:	AI W13
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Dr.-Ing. Harald Melcher		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Kenntnisse in der objektorientierten Programmierung

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- Plattformen und Programmiersprachen für mobile Applikationen
- Techniken zur Anpassung von Oberflächen an unterschiedlichste Anforderungen mobiler Endgeräte („responsive design“)
- Cross-Plattform-Aspekte von mobilen Anwendungen

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Oberflächen-für mobile Anwendung zu entwerfen und umzusetzen
- die lokale Datenhaltung sowie die Client-Server-Kommunikation zu entwickeln und zu implementieren
- Wirtschaftlichkeitsanalysen durchzuführen

Übergreifende Kompetenzen

Die Studierenden können

- komplette Anwendungen für gängige Betriebssysteme für mobile Endgeräte erstellen

Literaturhinweise:

Apple Developer Network: developer.apple.com

Android Developer Network: developer.android.com

Wird angeboten:

Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung/Projekt
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernergebnisse:

Die Studierenden sind in der Lage Applikationen für ein gängiges Betriebssystem für mobile Endgeräte erstellen zu können, wie z.B. ein Online-Spiel oder eine Marketing-Plattform. Dies schließt Icon- und Oberflächen-Entwurf, Bedien-Elemente auf verschiedenen Hardware-Plattformen, Client-Server-Kommunikation, lokale und entfernte Datenhaltung sowie Einstellen in einen entsprechenden App-Shop ein.

Die Studierenden verstehen Programmiersprachen für mobile Applikationen, sowie Plattformen und ihrer Bibliotheken. Sie erlernen die Möglichkeiten der Realisierung von Oberflächen und kennen deren Layout-Alternativen. Cross-Plattform-Aspekte, Barrierefreiheit und Location Based Services.

Bildung der Modulnote:

Klausur

Modulbeschreibung Data Mining

Schlüsselworte: Data Mining, Maschinelles Lernen , Visual Analytics, Data Science, Python

Zielgruppe:	1. oder 2.Semester AIM	Modulnummer:	AI W14
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Steffen Schober		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

grundlegende Programmierkenntnisse
mathematische Grundlagen

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- die Grundlagen des Data Mining: den Data Mining-Prozess, notwendige Daten-Vorverarbeitung
- Methoden der Visual Analytics: Scatter plots, Histogramme, Boxplots, Scatter plot matrix, Parallel Coordinates
- Verfahren des maschinellen Lernens
 - unüberwachtes Lernen: partitionierendes Clustering, hierarchisches Clustering
 - überwachtes Lernen: Klassifikations-/Regressionsverfahren, z.B. k-nearest neighbours, support vector machines, Entscheidungsbäume, Lineare Modelle)
- Grundlagen der Programmiersprache Python sowie des Python Data-Science Stacks (numpy, pandas, scikit-learn)

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- vorhandene Datensätze und Strukturen zu analysieren,
- geeignete Verfahren für bestimmte Probleme auszuwählen
- die erlernten Verfahren mit Hilfe der Programmiersprache Python einzusetzen

Übergreifende Kompetenzen

Die Studierenden können

- konkrete Probleme mit Hilfe von Data Mining Verfahren selbstständig lösen
- selbstständig Wissen aus Daten extrahieren

Inhalt:

Grundlagen des Data Mining:

- Data Mining-Prozess
- Datenvorverarbeitung
- Statistische Grundlagen

Maschinelles Lernen:

- Unüberwachtes Lernen
 - partitionierendes Clustering
 - hierarchisches Clustering
- Überwachtes Lernen
 - k-nearest neighbours, support vector machines, Entscheidungsbäume und Erweiterungen, Lineare Modelle

Visual Analytics:

- Scatter plots, Histogramme, Boxplots
- Scatter plot matrix, Parallel Coordinates

Praxis:

- Data Mining-Verfahren mit Python

Literaturhinweise:

Han, Jiawei: Data mining : concepts and techniques. - Amsterdam: Elsevier, 2012. - ISBN: 9780123814791

Hastie, Trevor J: The elements of statistical learning: data mining, inference, and prediction. - New York, NY: Springer, 2013. - ISBN: 9780387848570

Mazza, Riccardo: Introduction to information visualization. - London: Springer, 2009. - ISBN: 9781848002180

Wird angeboten:

Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen, Nachbereitung und Prüfungsvorbereitung
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernergebnisse:

Das Modul besteht aus theoretischer Vorlesung und einem praktischen Anteilen.

Nach erfolgreicher Absolvierung des Moduls sollen die Studierenden in der Lage sein, ausgewählte Data Mining-Verfahren selbstständig anzuwenden.

Die Studierenden haben Wissen erworben zu häufig verwendeten Data Mining-Verfahren und Big Data-Technologien und sollen in der Lage sein die geeignete Technologie für eine Aufgabenstellung zu wählen. Die Studierenden sollen Wissen erworben haben, das es ihnen erlaubt, selbstständig Wissen aus Daten zu extrahieren.

Bildung der Modulnote:

Klausur

Modulbeschreibung Mobile UX Design

Schlüsselworte: UX, UCD, Apps, Soziale Plattformen

Zielgruppe:	1. oder 2. Semester AIM	Modulnummer:	AI W15
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Astrid Beck		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Kenntnisse in der Programmierung

Modulziel – angestrebte Lernergebnisse:

Applikation für mobile Geräte zu entwickeln bedeutet neue Herausforderungen: Displaygröße, Kontextabhängigkeiten, neue Bedienkonzepte, unterschiedliche Außenbedingungen und die Forderung nach überzeugender User Experience (UX) müssen Entwickler meistern.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- das User Centered Design und insbesondere die Methoden User Research, Anforderungsermittlung, Prototyping sowie Usability Test für mobile Applikationen
- Tools und Templates

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Anforderungen, Konzeption und Design für mobile Endgeräte zu erstellen, wie z.B. ein Online-Spiel, eine Marketing-Plattform oder eine App im Automotive-Umfeld
- die unterschiedlichen Realisierungsmöglichkeiten zu unterscheiden und zu bewerten: Web, mobile Webseiten, Responsive Design, Native und Hybride Apps
- Tools und Templates in den Phasen des User Centered Design einzusetzen

Übergreifende Kompetenzen

Die Studierenden können

- Anforderungen des User Experience (UX) in Designs umsetzen
- soziale Plattformen in Apps einbinden

Inhalt:

- Statistiken: Systeme, Mobile Nutzer, Anwendungen
- Mobile Usability, Mobile UX
- User Centered design
- Projektplanung
- Marktanalyse Mitbewerber
- User Research, Kontextanalyse
- Anforderungsermittlung
- Prototyping
- Usability Test
- Realisierungsmöglichkeiten: Responsive Design, Native und Hybride Apps
- Style Guides und Normen
- Designprinzipien
- Graphische Gestaltungskonzepte, z.B. Material Design
- Zugänglichkeit/Barrierefreiheit

- Flexibles Design: HTML5, CSS3, u.a. Media Queries, flexible Font- und Bildgrößen
- Innovative Ansätze und Nutzung

Literaturhinweise:

ISO 9241-210

ISO 9241-11

Style Guides: Android, Apple, Windows

Wird angeboten:

Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:

Vorlesung und Projekt

Leistungskontrolle:

Projektarbeit: Bericht und Referat (20 Minuten)

Anteil Semesterwochenstunden:

4 SWS

Geschätzte studentische Arbeitszeit:

150 Stunden

Bildung der Modulnote:

Bericht und Referat (20 Minuten)

Modulbeschreibung Web Technologien

Schlüsselworte: Protokolle, Technologien, Anwendungen; Frameworks

Zielgruppe:	1. oder 2. Semester AIM	Modulnummer:	AI W16
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		30 h
	Selbststudium		120 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Prof. Dr. Andreas Rößler		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Grundkenntnisse Internettechnologien: HTML, CSS, JavaScript

Modulziel – angestrebte Lernergebnisse:

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- aktuelle Entwicklungen der Web-Protokolle, Server- und Client-Technologien
- aktuelle Web-UI-Frameworks

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- die Stärken und Schwächen von aktuellen Web-Technologien beurteilen
- aktuelle Web-Technologien in Projekten einsetzen

Übergreifende Kompetenzen

Die Studierenden können

- aktuelle Veröffentlichungen zum Stand der Technik von Web-Technologien finden, verstehen, einordnen, vermitteln und praktisch umsetzen
- eine eigene Webanwendung konzipieren und realisieren.

Inhalt:

- Web-Protokolle (http/2 und Nachfolger)
- Web-APIs (REST, GraphQL, ...)
- Web-Server-Technologien, z.B. Java/Jakarta EE, node.js, elixir, Go, ...
- Web-Frontend-Technologien, z.B. Angular, React, ...
- Technologien zur Verteilung, Qualitätssicherung, Steigerung der Performanz und Skalierbarkeit

Literaturhinweise:

Literatur wird von den betreuenden Dozenten für unterschiedlich Themen vorgeschlagen

Wird angeboten:

Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform: Seminar mit Bericht (max. 8 Seiten), Präsentation (1,5h) und Tutorial

Leistungskontrolle: Bericht und Präsentation

Anteil Semesterwochenstunden: 4 SWS

Geschätzte studentische Arbeitszeit: 150 Stunden

Lernergebnisse:

Die Studierenden sind in der Lage, sich in aktuelle Web-Technologien einzuarbeiten, diese zusammenzufassen, zu bewerten und zu vermitteln. Sie können Projekte mit aktuellen Web-Technologien systematisch aufsetzen und damit Anwendungen prototypisch erstellen.

Bildung der Modulnote:

Bericht und Präsentation

Modulbeschreibung Network Management

Schlüsselworte: Netzmanagement-Systeme, SNMP, NETCONF/YANG

Zielgruppe:	1. oder 2. Semester AIM	Modulnummer:	AI W18
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:	Englisch		
Modulverantwortung:	Prof. Dr.-Ing. Michael Scharf		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Kenntnisse in Rechnernetze, Kompetenzen zur Software-Entwicklung

Modulziel – angestrebte Lernergebnisse:

Die Studierenden lernen sowohl Grundlagen als auch Praxisbeispiele für Systeme, Protokolle und Verfahren zum Management von Kommunikationsnetzen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- Grundlegende Konzepte und Methoden für Konfiguration und Betrieb von Netzen
- Standards wie SNMP und NETCONF/YANG
- Wichtige Verfahren für das Netzmanagement und zur Netzüberwachung
- Fortgeschrittene Konzepte wie Telemetrie und Software-Defined Networking (SDN), insbesondere für die Automatisierung in Rechenzentren
- Beispiele für entsprechende Software-Applikationen

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Netztechnik wie beispielsweise Router zu konfigurieren und zu überwachen
- domänenspezifische Datenmodellierung anzuwenden (z.B. YANG)
- geeignete Software-Werkzeuge auszuwählen

Übergreifende Kompetenzen

Die Studierenden können

- Kommunikationsnetze in IT-Gesamtlösungen integrieren
- Software für die Automatisierung von Netzfunktionen entwerfen und einsetzen

Inhalt:

- Grundlagen des Managements von Kommunikationsnetzen
- Protokolle und Modellierungstechniken
- Netzmanagement-Systeme und Verfahren
- Anwendungen und Praxisbeispiele

Literaturhinweise:

Subramanian: Network Management: Principles and Practice, Pearson, 2011.
Stallings: SNMP, SNMPv2, SNMPv3, and RMON 1 and 2, Addison-Wesley, 1999.

Wird angeboten:

Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Projektarbeit
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernergebnisse:

Die Studierenden kennen die Konzepte von Netzmanagement und können diese auf einfache Beispiele anwenden.

Bildung der Modulnote:

Klausur

Modulbeschreibung Cloud Computing

Schlüsselworte: Cloud computing, Lights out computing, IaaS, PaaS, Container, Continuous Integration / Continuous Delivery, Cloud Functions

Zielgruppe:	1. oder 2. Semester AIM	Modulnummer:	AI W19
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		120 h
	Selbststudium		15 h
	Prüfungsvorbereitung		15 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Dipl.-Ing. Simon Moser		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Grundlagen der Informatik, Grundlegende Web-Technologien, Rechnernetze, Softwareentwicklung, Grundlegende Programmierkenntnisse

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, grundlegende Kenntnisse und Fertigkeiten im Umgang mit gängigen Techniken und Werkzeugen aus allen Bereichen des Cloud Computing – aus der Sicht von Anwendern, Bereitstellern und Anwendungsentwicklern - zu erwerben.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- Grundbegriffe des Cloud Computing: IaaS, PaaS, SaaS
- Grundelemente einer Cloud: Server, Netzwerke, etc.
- Grundprinzipien des Software Engineerings: Modularisierung, Abstraktion, etc.
- Grundlegende Web-Konzepte: HTTP, REST, etc.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Brücken zwischen physikalischer und virtueller Infrastruktur zu bilden.
- Zu verstehen wie Infrastruktur programmiert werden könnte und welche Möglichkeiten dies bietet.
- REST Web Services als Microservices in z.B. Docker Images zu deployen.

Übergreifende Kompetenzen

Die Studierenden können

- Wissen aus verschiedenen Domänen integrieren.
- in einem Team gemeinsam eine Aufgabe erfolgreich umzusetzen.
- verteilte Web-Architekturen mit Hilfe von Web Services konzipieren.

Inhalt:

„Cloud Computing“ ist ein umgangssprachlich sehr überladener Begriff, der eine Obermenge verschiedenster Technologien und Use Cases bildet. In der Vorlesung wird dem Studenten sowohl ein Überblick als auch ein detaillierter Einblick in all die Teilmengen des Themas vermittelt – beginnend von einer Begriffsklärung wird, anhand von praxisnahen Beispielen, jede Teilmenge des Themas „Cloud Computing“ beleuchtet, erläutert und selbst erforscht. Dabei werden im Verlauf verschiedene Sichten auf das Thema eingenommen – vom einfachen Nutzer eines Cloud Dienst über einen Cloud Betreiber bis hin zum Anwendungsentwickler von Cloud Diensten.

Literaturhinweise:

Thomas Erl: Cloud Computing: Concepts, Technology & Architecture (The Prentice Hall Service Technology Series from Thomas Erl)

Betsy Beyer, Chris Jones, Jennifer Petoff and Niall Richard Murphy : Site Reliability Engineering - <https://landing.google.com/sre/books/>

Wird angeboten:

Sommersemester

Teilgebiete und Leistungsnachweise:**Lehr- und Lernform:**

Vorlesung mit Übungen und Projektarbeit

Leistungskontrolle:

Klausur (90 Minuten) oder mündliche Prüfung (20 Minuten)

Anteil Semesterwochenstunden:

4 SWS

Geschätzte studentische Arbeitszeit:

150 Stunden

Lernergebnisse:

Die Studierenden

- verstehen die Bedeutung von Cloud Computing für die heutige Softwareentwicklung
- verstehen Virtualisierung und Software defined (Networking, Infrastruktur) über alle Ebenen hinweg
- verstehen Cloud Automatisierung und Lights-out-Computingkonzepte
- sind mit unterschiedlichen Entwicklungsmethoden und Prozessen hinsichtlich Cloud Computing vertraut
- können ein Konzept zur Migration von Anwendungen in die Cloud erarbeiten
- können eine Cloud-Anwendungsarchitektur entwickeln und ihre Implementierungsentscheidung detailliert begründen

Bildung der Modulnote:

Klausur oder mündliche Prüfung

Modulbeschreibung Advanced Data Models

Schlüsselworte: Data Models, NoSQL, Graph Database, Document Store, Semantic Web, Skill Based Engineering, Knowledge Graphs

Zielgruppe:	1. oder 2. Semester AIM	Modulnummer:	AI W20
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		50 h
	Selbststudium		100 h
Unterrichtssprache:	Deutsch oder Englisch		
Modulverantwortung:	Dr.-Ing. Jan R. Seyler		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Sehr gutes Abstraktionsvermögen, Programmierkenntnisse, Grundverständnis über Datenbanken, Kenntnisse über Modellierungssprachen (bspw. UML)

Modulziel – angestrebte Lernergebnisse:

Die Studierenden lernen sowohl Grundlagen als auch Praxisbeispiele für Systeme, Protokolle und Verfahren zum Management von Kommunikationsnetzen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- NoSQL Datenbankmodelle
- Semantische Webtechnologien wie Knowledge Graphen
- den Stand der Wissenschaft im Bereich Datenmodelle und Datenmanagement
- aktuelle, offene Problemstellungen aus Wissenschaft und Industrie im Bereich Datenmodellierung

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- zu vorliegenden Daten das richtige Datenmanagementsystem zu finden
- NoSQL Datenbanken aufzusetzen
- Queries in N1QL, SPARQL und Cypher formulieren.
- mit LaTeX wissenschaftliche Arbeiten verfassen.

Übergreifende Kompetenzen

Die Studierenden können

- Paper effizient lesen
- sich in neue Themen eigenständig einarbeiten
- ein wissenschaftliches Paper schreiben

Inhalt:

- Einführung in NoSQL Datenbankmodelle
- Vorstellung von Datenarten
- Das funktionale Modell
- Pitch & Verteilung der Themen
- Regelmäßige Zwischenmeetings während des Selbststudiums bei Festo
- Vorstellung der erarbeiteten Ergebnisse pro Gruppe in einem 30 minütigen Vortrag und mit Hilfe eines funktionalen Prototypen

Literaturhinweise:

- Sadalage, Pramod J., and Martin Fowler. "NoSQL distilled." AddisonWesley Professional (2012).
- Sullivan, Dan. NoSQL for mere mortals. Addison-Wesley Professional, 2015.
- Harrison, Guy. Next Generation Databases: NoSQL and Big Data. Apress, 2015.
- Yu, Liyang. Introduction to the semantic web and semantic web services. Chapman and Hall/CRC, 2007.
- Siegel, David. Pull: The power of the semantic web to transform your business. Penguin, 2009.
- West, Matthew. Developing high quality data models. Elsevier, 2011.
- Bubenko, Janis A. "From information algebra to enterprise modelling and ontologies—a historical perspective on modelling for information systems." Conceptual Modelling in Information Systems Engineering. Springer, Berlin, Heidelberg, 2007. 1-18.

Wird angeboten:

Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Seminar mit Bericht (max. 8 Seiten), Prototyp und Vortrag
Leistungskontrolle:	Referat (30 Minuten), Bericht
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernergebnisse:

Die Studierenden kennen verschiedene Datenarten und wissen mit welcher Datenbankart sich welche Daten besonders effizient speichern lassen. Des Weiteren haben die Studierenden einen Einblick in den aktuellen Stand der Technik und Forschung und offene Problemstellungen erhalten und hatten die Möglichkeit sich damit auseinanderzusetzen. Es werden das Abstraktionsvermögen und die Problemlösungsfähigkeit trainiert. Dieses Seminar gibt Studierenden die Chance sich mit den neusten Technologien und Problemstellungen im Bereich Datenmodelle zu beschäftigen. Die besten zwei Arbeiten zu einem Forschungsthema werden als Paper auf renommierten Konferenzen eingereicht.

Bildung der Modulnote:

Bericht 50%, Prototyp mit Referat 50%

Modulbeschreibung High Performance Computing

Schlüsselworte: Parallele Programmierung und Algorithmen, Performance

Zielgruppe:	1. oder 2. Semester AIM	Modulnummer:	AI W21
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		50 h
	Selbststudium		60 h
	Prüfungsvorbereitung		40 h
Unterrichtssprache:	Englisch oder Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Rainer Keller		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Programmierkenntnisse

Modulziel – angestrebte Lernergebnisse:

Die Studierenden lernen die Grundlagen der parallelen Programmierung.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen

- Grundlagen paralleler und nebenläufiger Programme:
 - Kommunikation über schnelle Netzwerke
 - Moderne Synchronisationsmechanismen
 - Fallstricke wie Deadlocks, Priority Inversion, etc.
- Werkzeuge, Technologien und Frameworks zur parallelen Programmierung

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Strategien zur Parallelisierung von Softwaresystemen beurteilen und auswählen
- Verschiedene Programmierschnittstellen anwenden, z.B. MPI, OpenCL und OpenACC für GPU-Programmierung
- Werkzeuge zum parallelen Debuggen anwenden
- Performance Analysen bei parallelen Systemen durchführen

Übergreifende Kompetenzen

Die Studierenden können

- Die Performance von Softwaresysteme mit Hilfe paralleler Programmierung verbessern
- Fehler in parallelen Programmen erkennen und beheben

Inhalt:

- Einführung in die parallele Programmierung
- Einblick parallele Programmierung mit Threads & OpenMP und nebenläufigem Code.
- Parallele Programmierung mit MPI und GASPI
- Parallele Programmierung für GPUs mittels OpenCL und OpenACC
- Effizienz von parallelen Algorithmen
- Performance Analyse Tools
- Verwendung von parallelen Debuggern

Literaturhinweise:

- MPI-Standard 3.1
- OpenCL 2.2
- Butenhof: Programming with POSIX Threads, Addison-Wesley
- Resch, Keller, Himmler, Krammer, Schulz: Tools for High Performance Computing, Springer-Verlag

Wird angeboten:

Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform: Vorlesung mit Übungen

Leistungskontrolle: Klausur (90 Min.)

Anteil Semesterwochenstunden: 4 SWS

Geschätzte studentische Arbeitszeit: 150 Stunden

Lernergebnisse:

Die Studierenden sind in der Lage, eigenen parallelen Code für CPU und GPU mit verschiedenen Programmierparadigmen für Systeme mit gemeinsamen und verteilten Speicher zu schreiben und vorhandenen Code zu parallelisieren. Sie haben eine Auswahl an Parallelisierungsparadigmen und einen Überblick und praktische Erfahrung in der Nutzung von Tools um diese anzuwenden.

Bildung der Modulnote:

Klausur

Modulbeschreibung Bewegungsplanung für autonome Systeme

Schlüsselworte: Robotik, Bewegungsplanung, Pfadplanung, Verhaltensplanung

Zielgruppe:	1. oder 2. Semester AIM	Modulnummer:	AI W22
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		120 h
	Selbststudium		15 h
	Prüfungsvorbereitung		15 h
Unterrichtssprache:	Deutsch		
Modulverantwortung:	Prof. Dr.-Ing. Thao Dang		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Lineare Algebra
Algorithmen und Datenstrukturen
Grundlagen Programmierkenntnisse in Python

Modulziel – angestrebte Lernergebnisse:

Die Studierenden beherrschen grundlegende Methoden der Bewegungsplanung in der Robotik. Im Speziellen verstehen sie die besonderen Anforderungen der Bewegungsplanung bei nicht-holonomen Robotern wie z.B. automatisierten Fahrzeugen. Sie sind in der Lage, Algorithmen der Bewegungsplanung für selbstfahrende Fahrzeuge selbst zu entwerfen und zu implementieren

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- Die Grundbegriffe der Bahn und Trajektorienplanung in der Robotik
- Bewegungsmodelle für selbstfahrende Fahrzeuge
- Grundlegende Algorithmen zur Bestimmung eines kostenoptimalen Pfades (diskrete Suchverfahren, probabilistische Verfahren, Methoden der Optimalsteuerung)
- Verhaltensgenerierung mit State Charts

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- Ein Bahn- oder Trajektorienplanungsproblem für mobile Roboter als diskretes oder kontinuierliches Optimierungsproblem mit Nebenbedingungen zu formulieren
- Lösungsverfahren für solche Optimierungsprobleme auszuwählen und zu implementieren

Übergreifende Kompetenzen

Die Studierenden können

- Planungs- und Regelungskomponenten für Roboter entwerfen und bewerten.

Inhalt:

- Übersicht Pfad- und Trajektorienplanung
- Einführung Robot Operating System (ROS)
- Verhaltensplanung für autonome Fahrzeuge
- Fahrzeugdynamik und Fahrzeugregelung (Ackermann, Einspurmodell, Deichselregler, Stanley)
- Grundlagen Bewegungsplanung in der Robotik (Grundbegriffe, Konfigurationsraum, Kollisionsberechnung, ...)
- Grundlegende Planungsverfahren (Bug Approaches, Potentialverfahren, Suchverfahren, A*, ...)

- Probabilistische Planungsverfahren (Probabilistic Roadmaps, RRTs, ...)
- Bewegungsplanung und Regelung mit Modellprädiktiven Reglern
- Einführung Reinforcement Learning
- Ausgewählte Themen der Absicherung, Rechnerhardware und Systemintegration im Hinblick auf die Verhaltenssteuerung automatisierter Fahrzeuge

Literaturhinweise:

- Howie Choset: Principles of Robot Motion – Theory, Algorithms and Implementation, MIT Press, 2005
- Steven M. Lavalle: Planning Algorithms, Cambridge University Press, 2006

Wird angeboten:

Wintersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen
Leistungskontrolle:	Klausur (90 Min.)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Lernergebnisse:

Die Studierenden können das Verhalten autonomer Fahrzeuge modellieren und mittels der verschiedenen Planungsverfahren und Vorhersagemodellen die Regelung dieser Fahrzeuge entwerfen und implementieren.

Bildung der Modulnote:

Klausur

Modulbeschreibung Information Security Management

Schlüsselworte: Informationssicherheit, Incident Response, Sicherheitmaßnahmen

Zielgruppe:	1. oder 2. Semester AIM	Modulnummer:	AI W23
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		120 h
	Selbststudium		15 h
	Prüfungsvorbereitung		15 h
Unterrichtssprache:	deutsch oder englisch		
Modulverantwortung:	Markus Schlittenhardt		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

Kenntnisse in Rechnernetzen, Netzwerke und Betriebssysteme

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, die Informationssicherheit in einer Organisation zu managen. Die Studierenden sind in der Lage zu verstehen, was Informationssicherheit innerhalb einer globalen Organisation bedeutet und wie diese aufgebaut werden kann. Zusätzlich werden die Studenten in die Lage versetzt zu verstehen, wie eine Organisation auf Sicherheitsvorfälle innerhalb der IT angemessen reagieren kann und welche Voraussetzungen dafür geschaffen werden müssen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die komplexen Auswirkungen von Sicherheitsanforderungen innerhalb der IT auf eine Organisation,
- die Funktionsweise eines Managementsystems für Informationssicherheit,
- die Auswirkungen von Risiken innerhalb der Informationstechnik auf eine Organisation und wie diese damit umgehen kann,
- die Voraussetzungen, die geschaffen werden müssen, dass eine Organisation auf Sicherheitsvorfälle innerhalb der IT reagieren kann,
- die Vorgehensweise auf Sicherheitsvorfälle angemessen zu reagieren und wie diese Vorfälle systematisch analysiert werden können.

Fertigkeiten – methodische Kompetenzen

Die Studierenden sind in der Lage

- auf Managementebene über Risiken innerhalb der Informationssicherheit zu diskutieren,
- auf Sicherheitsvorfälle innerhalb der Informationssicherheit zu reagieren sowie diese zu analysieren,
- Schwachstellen innerhalb einer Infrastruktur zu identifizieren und angemessen damit umzugehen.

Übergreifende Kompetenzen

Die Studierenden können

- komplexe Zusammenhänge und Auswirkungen der Informationssicherheit auf eine Organisation zu verstehen.

Inhalt:

- Einführung in die Informationssicherheit
- Technische Grundlagen für das Verständnis von Informationssicherheit
- Organisatorische Grundlagen für das Verständnis von Informationssicherheit
- Angriffe verstehen, analysieren und verhindern

Literaturhinweise:

- Thomas W. Harich; IT-Sicherheitsmanagement: Praxiswissen für IT Security Manager, mitp-Verlag, ISBN 978-3-95845-275-6, 2018
- André Domnick, et al; Informationssicherheit und Datenschutz: Handbuch für Praktiker und Begleitbuch zum T.I.S.P. dpunkt Verlag, ISBN 978-3-86490-596-4, 2019

Wird angeboten:

Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Projektarbeit
Leistungskontrolle:	KL 90 Minuten
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur

Course Description Network Security

Keywords: Secure Protocols, Authentication, Identity Management, Firewalls, Intrusion Detection

Audience:	1. oder 2. Semester AIM	Modul Number:	AI W24
Workload:	5 ECTS		150 h
divided into	Contact time		60 h
	Self-study		60 h
	Exam preparation		30 h
Course language:	English		
Modul director:	Prof. Dr.-Ing. Michael Scharf		
Vaild from:	01.03.2020		

Recommended requirements:

Understanding of computer networks, IT security and cryptography fundaments, basic programming skills

Desired learning outcomes of the module:

Students understand how to protect networks using both basic and advanced security methods.

Knowledge - professional competences

Students know:

- Network security objectives and basic attacks
- Security models for network protocols
- Cryptographic basics for network security protocols
- Security mechanisms on different network layers (e. g., PPP, IPsec, TLS, SSH)
- Authentication frameworks and identity management (e.g., OAuth, Kerberos, RADIUS)
- Basic protection solutions and devices (e.g., firewalls, VLAN, VPN, network monitoring, fail2ban)
- Advanced security mechanisms and algorithms (e.g., intrusion detection, honeypots)
- Anonymous communication

Skills - methodical competences

Students are able to

- Perform a security risk analysis for complex network deployments
- Select and implement network security methods
- Segment networks into security zones
- Design networks with regard to security
- Understand and use network security devices
- Understand anonymization techniques and their limitations

Comprehensive Competencies

Students be able to

- Deploy secure networked applications and IT services
- Leverage advanced concepts in network security

Contents:

- Network security goals, attacks and protection mechanisms
- Security mechanisms in the Internet (e.g., VLAN, IEEE 802.1X, IPsec, OpenVPN, TLS, SSH)
- Design and functions of network security protocols
- Authentication frameworks and identity management (e.g., Single-Sign-On, OAuth, Kerberos, PKI)
- Network attacks and counter-measures (e.g., firewalls, intrusion detection systems,)
- Advanced security solutions and research (e.g., intrusion detection, honeypots)
- Secure network operation and network monitoring
- Anonymous communication (Mixes, TOR)

Literature:

- W. Stallings: Network Security Essentials, Pearson Prentice Hall, 2007
- N. Ferguson, B. Schneier: Practical Cryptography John Wiley & Sons, 2003
- G. Schäfer, M. Roßberg: Netzsicherheit, 2. Auflage, dpunkt Verlag, 2014
- C. Eckert: IT-Sicherheit, Konzepte-Verfahren-Protokolle, Oldenbourg-Verlag, 2011
- R. Anderson: Security Engineering, Wiley, 2009
- B. Schneier: Applied Cryptography. Protocols, Algorithms, and Source Code in C. Wiley, New York 1996.

Offered:

Summer semester

Submodules and Assessment:

Type of instruction:	Lecture with exercises and project work
Type of assessment:	Exam (90 minutes)
Hours per week:	4 SWS
Estimated student workload:	150 Hours

Generation of the module grade:

Exam

Modulbeschreibung Secure Coding

Schlüsselworte: Code-Analyse, Schwachstellen in Codes, Cyber-Angriffe und Verteidigungsmaßnahmen

Zielgruppe:	1. oder 2. Semester AIM	Modulnummer:	AI W25
Arbeitsaufwand:	5 ECTS		150 h
Davon	Kontaktzeit		60 h
	Selbststudium		60 h
	Prüfungsvorbereitung		30 h
Unterrichtssprache:			
Modulverantwortung:	Prof. Dr. Dominik Schoop		
Stand:	01.03.2020		

Empfohlene Voraussetzungen:

- Kenntnisse in
- Programmiersprachen
 - IT Security

Modulziel – angestrebte Lernergebnisse:

Die Studierenden werden in die Lage versetzt, sichere Programme zu erstellen. Sie können Unsicherheiten im Code identifizieren und analysieren Sie können diese Schwachstellen beseitigen.

Kenntnisse – fachliche Kompetenzen

Die Studierenden kennen:

- die Grundursachen von Schwachstellen im Code und wie diese Schwachstellen ausgenutzt werden können.

Fertigkeiten – methodische Kompetenzen

Die Studierenden können:

- Schwachstellen im Code finden,
- Schwachstellen im Code beseitigen,
- Security Audits anwenden.

Übergreifende Kompetenzen

Die Studierenden sind in der Lage,

- Regeln für eine sichere Programmierung zu befolgen und das Sicherheitsniveau des Codes zu erhöhen.

Inhalt:

- Wichtigste Sicherheitslücken in Programmen
- Detaillierte Analyse der Datentypen und Speicherverwaltung
- Überläufe auf mehreren Ebenen
- Risiken bei der Datentyp-Konvertierung
- Zählen und Schleifen
- Sichere Ein- und Ausgänge, einschließlich Präprozessoreingänge
- Konzept der geringsten Privilegien und deren Anwendung
- Verschlüsselte temporäre Daten in Dateien und RAM
- Grundsätze der Code-Überprüfung und des Secure Software Engineering

Literaturhinweise:

- Jason Grembi, Secure Software Development: A Security Programmer's Guide Delmar Cengage Learning; 1 Edition, 2008
- Robert C. Seacord, The CERT C Secure Coding Standard Addison-Wesley Professional; 1 Edition , 2008
- Robert Seacord, Secure Coding in C and C++ Addison-Wesley Professional; 1 Edition, 2005
- Fred Long, Dhruv Mohindra, Robert C. Seacord, Dean F. Sutherland, David Svoboda, The CERT Oracle Secure Coding Standard for Java Addison-Wesley Professional; 1 edition, 2011

Wird angeboten:

Sommersemester

Teilgebiete und Leistungsnachweise:

Lehr- und Lernform:	Vorlesung mit Übungen und Projektarbeit
Leistungskontrolle:	Klausur (90 Minuten)
Anteil Semesterwochenstunden:	4 SWS
Geschätzte studentische Arbeitszeit:	150 Stunden

Bildung der Modulnote:

Klausur