

## **Fakultät Informationstechnik**

### **Modulhandbuch**

Master-Studiengang **Angewandte Informatik**  
mit Vertiefung

**Mobile Computing,**

**Autonome Systeme,**

**Business Intelligence**

## Inhaltsverzeichnis

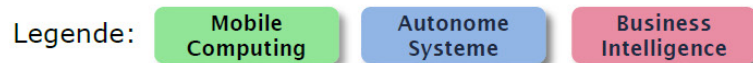
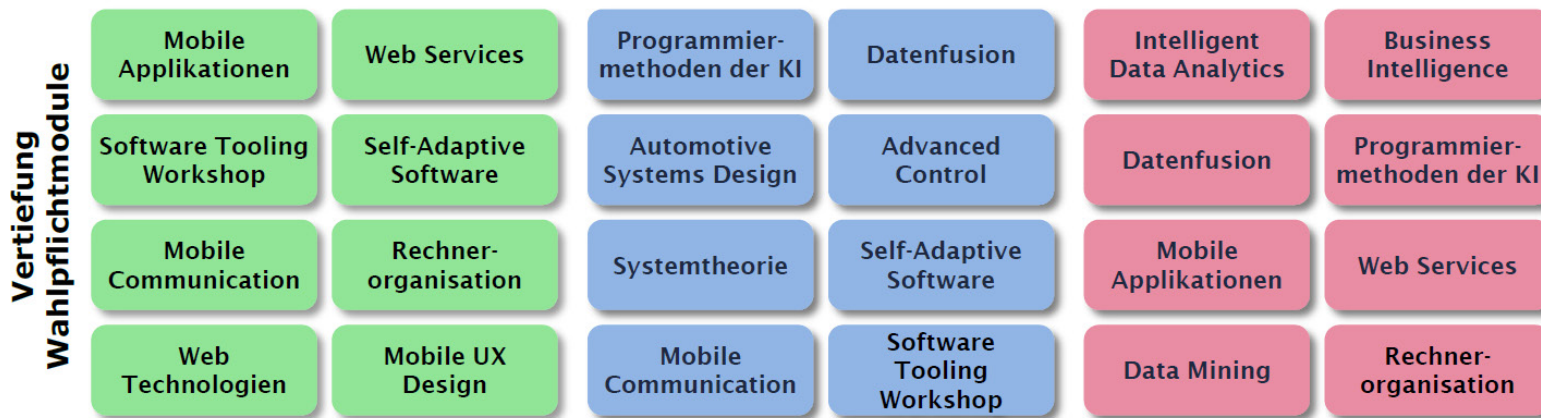
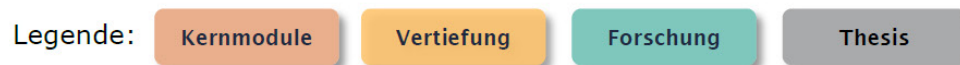
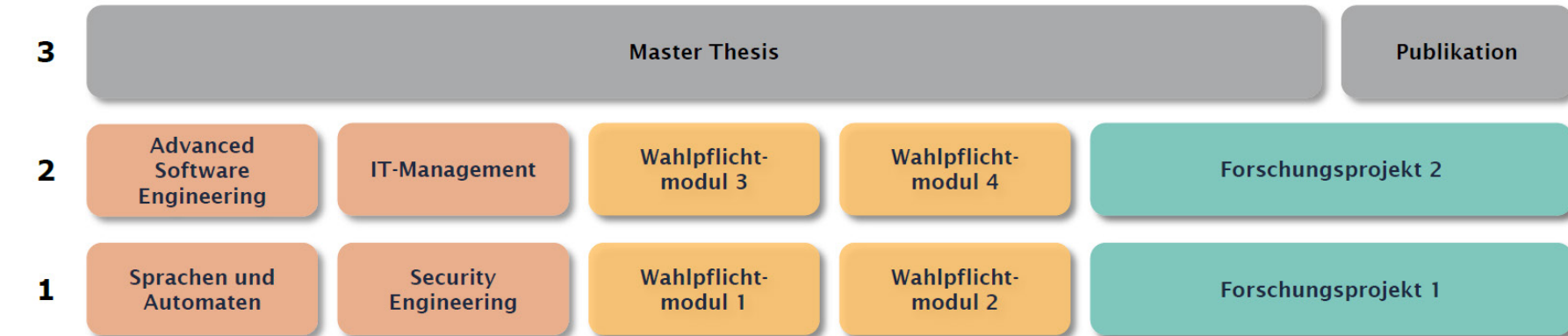
Modulnummer	Modul	Seite
	<b>Übersicht Modulplan</b>	<b>4</b>
<b>1. Semester</b>		
AI 101	Sprachen und Automaten	5
AI 102	Security Engineering	7
AI 103, AI 104	Wahlpflichtmodul 1 und 2	9
AI 105	Forschungsprojekt 1	10
<b>2. Semester</b>		
AI 201	Advanced Software Engineering	12
AI 202	IT Management	14
AI 203, AI 204	Wahlpflichtmodul 3 und 4	16
AI 205	Forschungsprojekt 2	17
<b>3. Semester</b>		
AI 301	Masterarbeit und Publikation	19
<b>Wahlpflichtmodule</b>		<b>21</b>
<b>Vertiefung Mobile Computing</b>		
AI W03	Mobile Communication	22
AI W09	Self-Adaptive Software	24
AI W10	Software Tooling Workshop	26
AI W11	Rechnerorganisation	28
AI W12	Web Services	30
AI W13	Mobile Applikationen	32
AI W15	Mobile UX Design	33
AI W16	Web Technologien	35
<b>Wahlpflichtmodule</b>		<b>37</b>
<b>Vertiefung Autonome Systeme</b>		
AI W03	Mobile Communication	38
AI W04	Datenfusion	40
AI W05	Automotive System Design	41
AI W06	Advanced Control	42
AI W07	Systemtheorie	43
AI W08	Programmiermethoden der KI	45
AI W09	Self-Adaptive Software	47
AI W10	Software Tooling Workshop	49
<b>Wahlpflichtmodule</b>		<b>51</b>
<b>Vertiefung Business Intelligence</b>		
AI W01	Intelligent Data Analytics	52
AI W02	Business Intelligence	54
AI W03	Datenfusion	56
AI W04	Programmiermethoden der KI	57
AI W08	Rechnerorganisation	59
AI W11	Web Services	61
AI W12	Mobile Applikationen	63

**Hinweis:**

Die genannten Voraussetzungen sind nicht zwingend, aber sehr hilfreich für das Verständnis der vermittelten Lerninhalte

## Übersicht Modulplan

Studienschwerpunkte **Mobile Computing, Autonome Systeme, Business Intelligence, IT-Entrepreneur**



## Modulbeschreibung Sprachen und Automaten

### Schlüsselworte: Theoretische Informatik

<b>Zielgruppe:</b>	<b>1. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI 101</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>60 h</b>
	<b>Selbststudium</b>		<b>60 h</b>
	<b>Prüfungsvorbereitung</b>		<b>30 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Dr. Heinrich Weber</b>		
<b>Stand:</b>	<b>29.06.2015</b>		

### Voraussetzungen:

Grundlagen der Informatik

### Gesamtziel:

Die Studierenden lernen den Aufbau formaler Sprachen und das Wissen, Probleme mit Formalen Sprachen und den ihnen zugeordneten Automaten zu beschreiben. Sie erwerben auch das Wissen zur Unterscheidung und Behandlung nichtdeterministischer und deterministischer Vorgänge anhand von Automatenmodellen. Die Studierenden lernen die Klassifizierung von Sprachen innerhalb der Chomsky-Hierarchie mit den zugeordneten Automatenmodellen. Die Studierenden erwerben die Kompetenz zur Analyse der Berechenbarkeit und zur Einschätzung der Komplexität von Problemen.

### Inhalt:

- Grammatiken
- Chomsky-Hierarchie
- Syntaxbäume
- Reguläre Sprachen
- Endliche, nicht endliche Automaten
- Reguläre Ausdrücke
- Kontextfreie Sprachen
- Kellerautomaten
- Kontextsensitive und allgemeine Sprachen
- Attributierte Grammatiken
- Beispiele und Übungen zur Generation und Akzeption

### Literaturhinweise:

Hoffmann v., D.: Theoretische Informatik; Hanser Verlag; 2011.  
Hedtstück, U.: Einführung in die Theoretische Informatik / Formale Sprachen und Automatentheorie, Oldenbourg Verlag; 2012.  
Erk, K; Priese, L: Theoretische Informatik / Eine umfassende Einführung; Springer Verlag; Berlin, Heidelberg, New York,... ; 2000.

### Wird angeboten:

in jedem Wintersemester

**Teilgebiete und Leistungsnachweise:**

<b>Lehr- und Lernform:</b>	Vorlesung
<b>Leistungskontrolle:</b>	Klausur (90 Minuten)
<b>Anteil Semesterwochenstunden:</b>	3 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	90 Stunden

**Lernergebnisse:**

Die Studierenden erwerben das notwendige theoretische Wissen, um Probleme mit Formalen Sprachen und Automaten zu beschreiben. Sie verfügen über die Fähigkeit, Methoden zur Beschreibung nichtdeterministischer und deterministischer Vorgänge anzuwenden. Sie können damit verschiedene Fragestellungen aus dem Bereich der Grammatiken und Automaten in Hinblick auf ihre Vollständigkeit, Abgeschlossenheit, Berechenbarkeit und Komplexität beurteilen. Sie erlernen die grundsätzlichen Möglichkeiten und Grenzen der Algorithmisierbarkeit kennen.

<b>Lehr- und Lernform:</b>	Übungen
<b>Leistungskontrolle:</b>	Testat
<b>Anteil Semesterwochenstunden:</b>	1 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	60 Stunden

**Lernergebnisse:**

Die Studierenden erlernen an Beispielen die formale Spezifikation von Sprachen. Sie haben das notwendige Wissen zur Einordnung von Sprachen in die bekannten Sprachklassen und kennen die ihnen jeweils zugeordneten Automatenmodelle. Sie erlernen die Dualität zwischen Akzeption und Generation. Sie kennen Verfahren zur Zustandsreduktion von Automaten und wenden diese in Übungen an. Sie lernen den grundsätzlichen Aufbau mehrschichtiger Grammatiken an Beispielen. Sie kennen die Grundlagen des Übersetzerbaus für höhere Programmiersprachen.

**Bildung der Modulnote:**

Klausur 100%

## Modulbeschreibung Security Engineering

**Schlüsselworte:** Security, Engineering, Software, Design, Standards

<b>Zielgruppe:</b>	<b>1. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI 102</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>60 h</b>
	<b>Selbststudium</b>		<b>60 h</b>
	<b>Prüfungsvorbereitung</b>		<b>30 h</b>
<b>Unterrichtssprache:</b>	<b>Englisch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Dr. Dominik Schoop</b>		
<b>Stand:</b>	<b>20.12.2014</b>		

**Voraussetzungen:**

Grundlagen des Softwareengineering

**Gesamtziel:**

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

**Inhalt:**

- Diskrete Mathematik
- Elliptische Kurven
- Faktorisierungsmethoden
- Moderne Verschlüsselungsverfahren
- Kryptographische Protokolle
- 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:**

<b>Lehr- und Lernform:</b>	Vorlesung
<b>Leistungskontrolle:</b>	Klausur (90 Minuten)
<b>Anteil Semesterwochenstunden:</b>	3 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	90 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.

<b>Lehr- und Lernform:</b>	Projektarbeit
<b>Leistungskontrolle:</b>	Testat
<b>Anteil Semesterwochenstunden:</b>	1 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	60 Stunden

**Lernergebnisse:**

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 100%



## Modulbeschreibung Wahlpflichtmodul 1 und 2

### Schlüsselworte: Vertiefung im eigenen Studienprofil

<b>Zielgruppe:</b>	<b>1. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI 103 AI 104</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>75 h</b>
	<b>Selbststudium</b>		<b>45 h</b>
	<b>Prüfungsvorbereitung</b>		<b>30 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch oder Englisch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Dr.-Ing. Andreas Rößler</b>		
<b>Stand:</b>	<b>01.07.2015</b>		

#### 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:

Abhängig vom gewählten Modul

#### Teilgebiete und Leistungsnachweise:

**Lehr- und Lernform:** Abhängig vom gewählten Modul

**Leistungskontrolle:** Abhängig vom gewählten Modul

**Anteil Semesterwochenstunden:** 5 ECTS

**Geschätzte studentische Arbeitszeit:** 150 Stunden

#### Lernergebnisse:

Die Studierenden verfügen über eine wissenschaftliche und fachliche Vertiefung im eigenen Studienprofil.

#### Bildung der Modulnote:

Abhängig vom gewählten Modul

## Modulbeschreibung Forschungsprojekt 1

**Schlüsselworte:** Wissenschaftliches Arbeiten im Team

<b>Zielgruppe:</b>	<b>1. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI 105</b>
<b>Arbeitsaufwand:</b>	<b>10 ECTS</b>		<b>300 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>40 h</b>
	<b>Selbststudium</b>		<b>260 h</b>
	<b>Prüfungsvorbereitung</b>		<b>0 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch oder Englisch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Dr.-Ing. Andreas Rößler</b>		
<b>Stand:</b>	<b>27.12.2014</b>		

**Voraussetzungen:**

Anwendung der Methoden des Softwareentwicklung, Kenntnisse in der gewählten Vertiefungsrichtung

**Gesamtziel:**

Die Studierenden sind in der Lage:

- wissenschaftliche Methoden anzuwenden,
- wissenschaftliche Projekte zu planen und durchzuführen,
- Problemstellungen aus der Forschung oder aus der Industrie im Team innerhalb einer vorgegebenen Frist zu lösen,
- Lösungsansätze (Stand der Forschung) zu recherchieren und zu verstehen,
- vorhandene Lösungsansätze einzusetzen und eigene Lösungsansätze zu entwickeln,
- Lösungen und Ergebnisse zu evaluieren
- die Vorgehensweise und die Ergebnisse zu dokumentieren

**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 Zwischen- bzw. Endergebnisse vor.

**Literaturhinweise:**

Abhängig von der gewählten Problemstellung

**Wird angeboten:**

Sommer- und Wintersemester

**Teilgebiete und Leistungsnachweise:**

<b>Lehr- und Lernform:</b>	Projektarbeit mit Bericht und Präsentation
<b>Leistungskontrolle:</b>	Schriftlicher Bericht und Referat (20 Minuten)
<b>Anteil Semesterwochenstunden:</b>	10 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	300 Stunden

**Lernergebnisse:**

Die Studierenden kennen wissenschaftliche Methoden und können diese anwenden. Sie haben Kenntnisse im Projektmanagement und können wissenschaftliche Projekte planen und durchführen. Die Studierenden sind in der Lage, die in den Kern- und Vertiefungsfächern erworbenen Kenntnisse zur Lösung von Problemen aus dem Bereich der Forschung einzusetzen. Sie verfügen über die Kompetenzen, selbständig neue Kenntnisse in der Informatik zu gewinnen und neue Verfahren zu entwickeln. Dazu können sie Wissen aus verschiedenen Domänen integrieren. Die Studierenden haben soziale Kompetenzen erworben, die sie in Lage versetzen in einem Team gemeinsam eine Aufgabe erfolgreich umzusetzen.

**Bildung der Modulnote:**

Projekt: Referat und Dokumentation

## Modulbeschreibung Advanced Software Engineering

**Schlüsselworte:** Software Engineering, Metriken, Empirie

<b>Zielgruppe:</b>	<b>2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI 201</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>30 h</b>
	<b>Selbststudium</b>		<b>120 h</b>
	<b>Prüfungsvorbereitung</b>		<b>0 h</b>
<b>Unterrichtssprache:</b>	<b>Englisch oder Deutsch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Hans-Gerhard Groß</b>		
<b>Stand:</b>	<b>10.12.2014</b>		

**Voraussetzungen:**

Grundlagen des Softwareengineering

**Gesamtziel:**

Die Studierenden sind in der Lage:

- existierende Software Metriken anzuwenden und deren Aussagekraft abzuschätzen; neue Software Metriken zu entwickeln und zu validieren
- Software Engineering Probleme als Suchprobleme zu formulieren und mit geeigneten Suchverfahren zu lösen
- empirische Studien im Bereich des Software Engineering zu planen, durchzuführen und zu evaluieren
- relevante Forschungs-Fragen im Software Engineering zu erkennen und zu formulieren
- einfache Forschungsthemen des Software Engineering zu bearbeiten, Lösungen zu entwickeln und zu dokumentieren

**Inhalt:**

- Software Measurement und Software Metrics
- Search-based Software Engineering, Anwendung von Suchheuristiken zur Lösung von SWE Problemen
- Experimental Software Engineering
- Weitere selbst-definierte Themen im Bereich des Software-Engineering (Seminararbeit)

**Literaturhinweise:**

Somerville: Software Engineering.  
Ludewig, Lichter: Software Engineering. dpunkt.verlag.  
Shepperd, Ince: Derivation and Evaluation of Software Metrics, Claredon Press.  
Wohlin, et. Al: Experimentation in Software Engineering, Springer.  
Proceedings of the International Symposium on Search-Based Software Engineering (www.sbse.org).

**Wird angeboten:**

in jedem Sommersemester

**Teilgebiete und Leistungsnachweise:**

<b>Lehr- und Lernform:</b>	Seminar
<b>Leistungskontrolle:</b>	Individuelle Seminararbeit
<b>Anteil Semesterwochenstunden:</b>	1 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	30 Stunden

**Lernergebnisse:**

Die Studierenden sind in der Lage ein aktuelles Thema aus dem Bereich des Software Engineering selbständig aufzuarbeiten und vorzutragen.

<b>Lehr- und Lernform:</b>	Projektarbeit
<b>Leistungskontrolle:</b>	Abschlusspräsentation
<b>Anteil Semesterwochenstunden:</b>	2 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	60 Stunden

**Lernergebnisse:**

Die Studierenden sind in der Lage:

- empirische Studien im Bereich des Software Engineering zu planen, durchzuführen und zu evaluieren
- relevante Forschungs-Fragen im Software Engineering zu erkennen und zu formulieren einfache Forschungsthemen des Software Engineering zu bearbeiten, Lösungen zu entwickeln und zu dokumentieren

<b>Lehr- und Lernform:</b>	Online-Kurs, Selbststudium
<b>Leistungskontrolle:</b>	Individuelle Seminararbeit
<b>Anteil Semesterwochenstunden:</b>	0 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	30 Stunden

**Lernergebnisse:**

Die Studierenden sind in der Lage:

- existierende Software Metriken anzuwenden und deren Aussagekraft abzuschätzen; neue Software Metriken zu entwickeln und zu validieren
- Software Engineering Probleme als Suchprobleme zu formulieren und mit geeigneten Suchverfahren zu lösen

**Bildung der Modulnote:**

Schriftliche Arbeit (S) und Referat (RE)

## Modulbeschreibung IT Management

### Schlüsselworte: IT Management

<b>Zielgruppe:</b>	<b>2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI 202</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>60 h</b>
	<b>Selbststudium</b>		<b>60 h</b>
	<b>Prüfungsvorbereitung</b>		<b>30 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Dr. Thomas Rodach</b>		
<b>Stand:</b>	<b>02.07.2015</b>		

### Voraussetzungen:

Grundkenntnisse in Betriebswirtschaft sowie von Informationssystemen in Unternehmen

### Gesamtziel:

Dieses Modul behandelt die Managementaufgaben in der IT, die im Rahmen von Führungsfunktionen wahrzunehmen sind.

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

### 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:**

<b>Lehr- und Lernform:</b>	Vorlesung
<b>Leistungskontrolle:</b>	Mündliche Prüfung (20 Minuten)
<b>Anteil Semesterwochenstunden:</b>	2 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	60 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

<b>Lehr- und Lernform:</b>	Fallstudien / Projekt
<b>Leistungskontrolle:</b>	Referat
<b>Anteil Semesterwochenstunden:</b>	2 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	90 Stunden

**Lernergebnisse:**

Die Studierenden

- 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 Prüfungsleistung 100%

## Modulbeschreibung Wahlpflichtmodul 3 und 4

### Schlüsselworte: Vertiefung im eigenen Studienprofil

<b>Zielgruppe:</b>	<b>1. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI 203 AI 204</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>75 h</b>
	<b>Selbststudium</b>		<b>45 h</b>
	<b>Prüfungsvorbereitung</b>		<b>30 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch oder Englisch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Dr.-Ing. Andreas Rößler</b>		
<b>Stand:</b>	<b>01.07.2015</b>		

#### 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:

Abhängig vom gewählten Modul

#### Teilgebiete und Leistungsnachweise:

**Lehr- und Lernform:** Abhängig vom gewählten Modul

**Leistungskontrolle:** Abhängig vom gewählten Modul

**Anteil Semesterwochenstunden:** 5 ECTS

**Geschätzte studentische Arbeitszeit:** 150 Stunden

#### Lernergebnisse:

Die Studierenden verfügen über eine wissenschaftliche und fachliche Vertiefung im eigenen Studienprofil.

#### Bildung der Modulnote:

Abhängig vom gewählten Modul



## Modulbeschreibung Forschungsprojekt 2

**Schlüsselworte:** Wissenschaftliches Arbeiten im Team

<b>Zielgruppe:</b>	<b>2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI 205</b>
<b>Arbeitsaufwand:</b>	<b>10 ECTS</b>		<b>300 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>40 h</b>
	<b>Selbststudium</b>		<b>260 h</b>
	<b>Prüfungsvorbereitung</b>		<b>0 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch oder Englisch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Dr.-Ing. Andreas Rößler</b>		
<b>Stand:</b>	<b>27.12.2014</b>		

**Voraussetzungen:**

Anwendung der Methoden des Softwareentwicklung, Kenntnisse in der gewählten Vertiefungsrichtung

**Gesamtziel:**

Die Studierenden sind in der Lage:

- wissenschaftliche Methoden anzuwenden,
- wissenschaftliche Projekte zu planen und durchzuführen,
- Problemstellungen aus der Forschung oder aus der Industrie im Team innerhalb einer vorgegebenen Frist zu lösen,
- Lösungsansätze (Stand der Forschung) zu recherchieren und zu verstehen,
- vorhandene Lösungsansätze einzusetzen und eigene Lösungsansätze zu entwickeln,
- Lösungen und Ergebnisse zu evaluieren
- die Vorgehensweise und die Ergebnisse zu dokumentieren

**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 Zwischen- bzw. Endergebnisse vor.

**Literaturhinweise:**

Abhängig von der gewählten Problemstellung

**Wird angeboten:**

Sommer- und Wintersemester

**Teilgebiete und Leistungsnachweise:**

<b>Lehr- und Lernform:</b>	Projektarbeit mit Bericht und Präsentation
<b>Leistungskontrolle:</b>	Schriftlicher Bericht und Referat (20 Minuten)
<b>Anteil Semesterwochenstunden:</b>	10 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	300 Stunden

**Lernergebnisse:**

Die Studierenden kennen wissenschaftliche Methoden und können diese anwenden. Sie haben Kenntnisse im Projektmanagement und können wissenschaftliche Projekte planen und durchführen. Die Studierenden sind in der Lage, die in den Kern- und Vertiefungsfächern erworbenen Kenntnisse zur Lösung von Problemen aus dem Bereich der Forschung einzusetzen. Sie verfügen über die Kompetenzen, selbständig neue Kenntnisse in der Informatik zu gewinnen und neue Verfahren zu entwickeln. Dazu können sie Wissen aus verschiedenen Domänen integrieren. Die Studierenden haben soziale Kompetenzen erworben, die sie in Lage versetzen in einem Team gemeinsam eine Aufgabe erfolgreich umzusetzen.

**Bildung der Modulnote:**

Projekt: Referat und Dokumentation

## Modulbeschreibung Masterarbeit und Publikation

### Schlüsselworte: Selbständiges wissenschaftliches Arbeiten

<b>Zielgruppe:</b>	<b>3. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI 301</b>
<b>Arbeitsaufwand:</b>	<b>30 ECTS</b>		<b>900 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>80 h</b>
	<b>Selbststudium</b>		<b>740 h</b>
	<b>Prüfungsvorbereitung</b>		<b>80 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch oder Englisch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Dr.-Ing. Andreas Rößler</b>		
<b>Stand:</b>	<b>12.07.2015</b>		

### Voraussetzungen:

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

### Gesamtziel:

Die Studierenden sind in der Lage:

- eine Problemstellung der Informatik selbständig, 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
- Präsentation und Verteidigung

### Literaturhinweise:

Balzert, H., M. Schröder und C. Schäfer, 2014. Wissenschaftliches Arbeiten. Ethik, Inhalt & Form wiss. Arbeiten, Handwerkszeug, Quellen, Projektmanagement, Präsentation. 2. Nachdr. Dortmund: W3L-Verl.

Ludewig, J. und H. Lichter, 2013. Software Engineering. Grundlagen Menschen Prozesse Techniken. 3., korr. Aufl. Heidelberg: dpunkt-Verl.

Fachliteratur abhängig von der gewählten Problemstellung

### Wird angeboten:

in jedem Semester

**Teilgebiete und Leistungsnachweise:**

<b>Modulteil:</b>	Masterarbeit
<b>Lehr- und Lernform:</b>	Wissenschaftliche Arbeit mit Bericht und Verteidigung
<b>Leistungskontrolle:</b>	Schriftlicher Bericht und 20 Minuten Referat
<b>Anteil Semesterwochenstunden:</b>	25 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	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 selbständigen Projekt- und Zeitplanung und haben Verantwortungsbewusstsein in Bezug auf wissenschaftliche Qualitätskriterien wie Überprüfbarkeit und Verlässlichkeit von wissenschaftlichen Ergebnissen erworben.

<b>Modulteil:</b>	Publikation
<b>Lehr- und Lernform:</b>	Verfassen einer wissenschaftlichen Publikation
<b>Leistungskontrolle:</b>	Veröffentlichungsreifer Text
<b>Anteil Semesterwochenstunden:</b>	5 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	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, selbständig für die Qualitätssicherung der Publikation zu sorgen und diese fristgerecht einzureichen.

**Bildung der Modulnote:**

Gemittelte Note aus Bericht (Faktor 4), Verteidigung (Faktor 1) und Publikation (Faktor 1)

# Wahlpflichtmodule

## Vertiefung **Mobile Computing**

### Inhaltsverzeichnis

<b>Modulnummer</b>	<b>Modul</b>	<b>Seite</b>
AI W03	<b>Mobile Communication</b>	22
AI W09	<b>Self-Adaptive Software</b>	24
AI W10	<b>Software Tooling Workshop</b>	26
AI W11	<b>Rechnerorganisation</b>	28
AI W12	<b>Web Services</b>	30
AI W13	<b>Mobile Applikationen</b>	32
AI W15	<b>Mobile UX Design</b>	33
AI W16	<b>Web Technologien</b>	35

## Modulbeschreibung Mobile Communication

**Schlüsselworte:** Automotive, Communication, Safety, Security

<b>Zielgruppe:</b>	1. oder 2. Semester AIM	<b>Modulnummer:</b>	AI W03
<b>Arbeitsaufwand:</b>	5 ECTS		150 h
<b>Davon</b>	<b>Kontaktzeit</b>		50 h
	<b>Selbststudium</b>		100 h
	<b>Prüfungsvorbereitung</b>		0 h
<b>Unterrichtssprache:</b>	Englisch		
<b>Modulverantwortung:</b>	Prof. Dr.-Ing. Harald Melcher, Prof. Dr. Dominik Schoop, Prof. Dr.-Ing. Martin Zieher		
<b>Stand:</b>	20.12.2014		

### Voraussetzungen:

Grundlagen Kommunikationstechnik  
Module IT-Security Engineering and Advanced Software Engineering

### Gesamtziel:

Die Studierenden sind in der Lage:

- die Ziele von ITS und VANETs zu erklären
- die Architektur und Technologie von VANETs zu erklären
- die Sicherheit von VANETs einzuschätzen und Sicherheitsmaßnahmen vorzuschlagen
- einfache Car2Infrastructure- und Car2-Car-Applikationen zu implementieren

### Inhalt:

- Überblick Intelligent Transportation Systems (ITS)
- 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, ...)
- Zusammenhang Safety und Security
- Angriffe in VANETs
- Authentizität, Anonymität, Integrität, Vertraulichkeit und Missbrauchserkennung in VANETs
- pseudonyme PKIs

### Literaturhinweise:

Veröffentlichungen des Car-2-Car Communication Consortium (C2CCC), [www.car-2-car.org](http://www.car-2-car.org)  
Veröffentlichungen des Projektes „Sichere Intelligente Mobilität Testfeld Deutschland (simTD)“, [www.simtd.de](http://www.simtd.de)  
Veröffentlichungen des Projektes „Secure Vehicular Communication (SeVeCom)“, [www.sevecom.org](http://www.sevecom.org)  
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:

Einmal pro Jahr (Sommer- oder Wintersemester)

**Teilgebiete und Leistungsnachweise:**

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

**Lernergebnisse:**

Die Studierenden sind in der Lage:

- die Ziele von ITS und VANETs zu erklären
- die Architektur und Technologie von VANETs zu erklären

<b>Lehr- und Lernform:</b>	Projektarbeit (Labor)
<b>Leistungskontrolle:</b>	Abschlusspräsentation
<b>Anteil Semesterwochenstunden:</b>	0,5 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	90 Stunden

**Lernergebnisse:**

Die Studierenden sind in der Lage:

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

**Bildung der Modulnote:**

Klausur, unbenotetes Testat und unbenoteter Bericht

## Modulbeschreibung Self-Adaptive Software

### Schlüsselworte: Software Self-\* Techniques

<b>Zielgruppe:</b>	<b>1. oder 2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI W09</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>30 h</b>
	<b>Selbststudium</b>		<b>120 h</b>
	<b>Prüfungsvorbereitung</b>		<b>0 h</b>
<b>Unterrichtssprache:</b>	<b>Englisch oder Deutsch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Hans-Gerhard Groß</b>		
<b>Stand:</b>	<b>16.12.2014</b>		

### Voraussetzungen:

Grundlagen des Softwareengineering, Software Architektur, Distributed Systems

### Gesamtziel:

Die Studierenden sind in der Lage:

- Geeignete Techniken zur Realisierung von adaptiven Systemen zu evaluieren, auszuwählen und integrieren.
- System Monitoring zu entwickeln und effizient einzusetzen.
- Diagnoseverfahren zu entwickeln und effizient einzusetzen.
- Neue Methoden und Techniken zu entwickeln, die eine automatische Software Adaption ermöglichen.

### Inhalt:

- Individuelle Projektarbeit zum Thema Software Self-\* Techniques.
- Adaptive Software Platforms, Software Monitoring and Diagnosis, Software Healing
- 24-7 Uptime Requirements of Software Systems.
- Entwicklung eines Prototypen, der die gewählte Self-\* Technique realisiert.

### Literaturhinweise:

Cheng, B.H.C., de Lemos, R., Inverardi, P., Magee, J. (Eds.) Software Engineering for Self-Adaptive Systems, Springer LNCS 5525, 2009.

Individuelle Literatursuche entspr. des gewählten Themas (Literature Search).

### Wird angeboten:

Einmal pro Jahr (Sommer- oder Wintersemester)

### Teilgebiete und Leistungsnachweise:

<b>Lehr- und Lernform:</b>	Seminar
<b>Leistungskontrolle:</b>	Seminararbeit
<b>Anteil Semesterwochenstunden:</b>	1 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	30 Stunden

### Lernergebnisse:

Die Studierenden sind in der Lage aktuelle Themen im Bereich Software Self-\* Techniques selbstständig zu recherchieren und zu präsentieren.



**Lehr- und Lernform:** Projekt  
**Leistungskontrolle:** Projektarbeit  
**Anteil Semesterwochenstunden:** 2 SWS  
**Geschätzte studentische Arbeitszeit:** 60 Stunden

**Lernergebnisse:**

Die Studierenden sind in der Lage:

- Geeignete Techniken zur Realisierung von adaptiven Systemen zu evaluieren, auszuwählen und integrieren.
- System Monitoring zu entwickeln und effizient einzusetzen.
- Diagnoseverfahren zu entwickeln und effizient einzusetzen.
- Neue Methoden und Techniken zu entwickeln, die eine automatische Software Adaption ermöglichen..

**Lehr- und Lernform:** Online-Kurs, Selbststudium  
**Leistungskontrolle:** Hausarbeit  
**Anteil Semesterwochenstunden:** 1 SWS  
**Geschätzte studentische Arbeitszeit:** 30 Stunden

**Lernergebnisse:**

Die Studierenden besitzen Kenntnisse über die Methoden von adaptiven Software Systemen.

**Bildung der Modulnote:**

Klausur

## Modulbeschreibung Software Tooling Workshop

**Schlüsselworte:** Software Tools, Qualität, DSL

<b>Zielgruppe:</b>	<b>1. oder 2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI W10</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>30 h</b>
	<b>Selbststudium</b>		<b>120 h</b>
	<b>Prüfungsvorbereitung</b>		<b>0 h</b>
<b>Unterrichtssprache:</b>	<b>Englisch oder Deutsch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Hans-Gerhard Groß</b>		
<b>Stand:</b>	<b>16.12.2014</b>		

**Voraussetzungen:**

Grundlagen des Softwareengineering, Software Architektur

**Gesamtziel:**

Die Studierenden sind in der Lage:

- Existierende Theorien des Software Quality Management in entsprechende Tools zur Messung von Qualität umzusetzen und prototypisch zu realisieren.
- Tools für ausgewählte Phasen des Software-Lebenszyklus zu entwickeln und zu evaluieren.
- Moderne Sprachen der 4. Generation effizient für die schnelle Entwicklung von Software-Tools anzuwenden.

**Inhalt:**

- Individuelle Projektarbeit zum Thema Software Tools.
- Entwicklung eines Tool-Prototyps.
- Evaluierung der Technologien plus Evaluierung des Tools mittels Fallstudie.
- Dokumentation der zugrunde liegenden Technologien, des Tools, und der Evaluierung

**Literaturhinweise:**

Individuelle Literatursuche entspr. des gewählten Themas (Literature Search).

**Wird angeboten:**

Einmal pro Jahr (Sommer- oder Wintersemester)

**Teilgebiete und Leistungsnachweise:**

<b>Lehr- und Lernform:</b>	Seminar
<b>Leistungskontrolle:</b>	Seminararbeit
<b>Anteil Semesterwochenstunden:</b>	1 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	30 Stunden

**Lernergebnisse:**

Die Studierenden sind in der Lage aktuelle Themen im Bereich Software Tooling selbständig zu recherchieren und zu präsentieren.

<b>Lehr- und Lernform:</b>	Projekt
<b>Leistungskontrolle:</b>	Projektarbeit
<b>Anteil Semesterwochenstunden:</b>	3 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	90 Stunden

**Lernergebnisse:**

Die Studierenden sind in der Lage:

- Existierende Theorien des Software Quality Management in entsprechende Tools zur Messung von Qualität umzusetzen und prototypisch zu realisieren.
- Tools für ausgewählte Phasen des Software-Lebenszyklus zu entwickeln und zu evaluieren.
- Moderne Sprachen der 4. Generation effizient für die schnelle Entwicklung von Software-Tools anzuwenden

**Bildung der Modulnote:**

Bericht und Referat benotet, unbenotetes Testat

## Modulbeschreibung Rechnerorganisation

### Schlüsselworte: Praktische Informatik

<b>Zielgruppe:</b>	<b>1. oder 2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI W11</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>60 h</b>
	<b>Selbststudium</b>		<b>60 h</b>
	<b>Prüfungsvorbereitung</b>		<b>30 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Dr. Heinrich Weber</b>		
<b>Stand:</b>	<b>02.01.2015</b>		

### Voraussetzungen:

Vorlesung Betriebssysteme, eine Programmiersprache

### Gesamtziel:

Die Studierenden lernen die erweiterte Systemprogrammierung und Systemkonfiguration zur Erstellung und zum Betrieb eines verteilten, heterogenen Rechnernetzes, wie es in Rechenzentren oder von Dienstleistungs Providern vorgehalten wird. Dies umfasst die Integration von Hardwarekomponenten und Treibersoftware ebenso wie die Einrichtung und den Betrieb der benötigten Rechnerdienste (z.B. DHCP/BOOTP, BIND, NFS, NIS, LDAP, HTTPD, SAMBA, etc.) In die Integration der Dienste werden Verfahren zur Abwehr von Rechnerangriffen und Methoden der Computerforensik mit einbezogen.

### Inhalt:

- Rechner- und Betriebssystem-Bootvorgang
- Einbinden spezieller Rechner-Hardware, Treibermodule
- System-V und andere Upstart-Services
- Netzwerkintegration und einfache Netzwerkservices
- Erweiterte Netzwerkdienste
- Userverwaltung und Authentifizierung in großen Netzen
- Serverdienste für größere Netzwerke
- Windows, Unix und Mac in einem Netzwerk
- Sicherheitsaspekte und Abwehr von Angriffen

### Literaturhinweise:

Scott, M. Mitchell, E.L.: Linux System Security, Prentice Hall, 2-nd Ed., 2002  
Kettner, M.: Fehlerdiagnose und Problembhebung unter Linux, Suse Press, 2004

### Wird angeboten:

Wintersemester

**Teilgebiete und Leistungsnachweise:**

<b>Lehr- und Lernform:</b>	Vorlesung
<b>Leistungskontrolle:</b>	Klausur (90 Minuten)
<b>Anteil Semesterwochenstunden:</b>	3 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	90 Stunden

**Lernergebnisse:**

Die Studierenden kennen die erweiterte Systemprogrammierung und Systemkonfiguration zur Erstellung und zum Betrieb eines verteilten, heterogenen Rechnernetzes, wie es in Rechenzentren oder von Dienstleistungs Providern vorgehalten wird. Dies umfasst die Integration von Hardwarekomponenten und Treibersoftware ebenso wie die Einrichtung und den Betrieb der benötigten Rechnerdienste (z.B. DHCP/BOOTP, BIND, NFS, NIS, LDAP, HTTPD, SAMBA, etc.) In die Integration der Dienste werden Verfahren zur Abwehr von Rechnerangriffen und Methoden der Computerforensik mit einbezogen.

<b>Lehr- und Lernform:</b>	Übungen (Labor)
<b>Leistungskontrolle:</b>	Testat
<b>Anteil Semesterwochenstunden:</b>	1 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	60 Stunden

**Lernergebnisse:**

Die Studierenden sind in der Lage ein verteiltes, heterogenes Rechnernetz zu konfigurieren und in Betrieb zu nehmen.

**Bildung der Modulnote:**

Klausur, unbenotetes Testat

## Modulbeschreibung Web Services

**Schlüsselworte:** Serviceorientierte Architekturen, Web-Anwendungen

<b>Zielgruppe:</b>	<b>1. oder 2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI W12</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>60 h</b>
	<b>Selbststudium</b>		<b>60 h</b>
	<b>Prüfungsvorbereitung</b>		<b>30 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch oder Englisch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Dr.-Ing. Andreas Rößler</b>		
<b>Stand:</b>	<b>30.12.2014</b>		

**Voraussetzungen:**

Grundlegende Web-Technologien, Rechnernetze, Softwareentwicklung

**Gesamtziel:**

Die Studierenden können verteilte Systeme auf der Basis von Web Services aufbauen.

**Inhalt:**

- Grundlagen der Web Services
- Serviceorientierte Architekturen
- Verteilte Anwendungen
- Middleware
- SOAP, SOAP over HTTP
- Web Services Description Language
- UDDI: Universal Description, Discovery and Integration
- Sichere Web Services

**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:**

Einmal pro Jahr (Sommer- oder Wintersemester)

**Teilgebiete und Leistungsnachweise:**

<b>Lehr- und Lernform:</b>	Vorlesung / Seminar
<b>Leistungskontrolle:</b>	Präsentation (30 Minuten)
<b>Anteil Semesterwochenstunden:</b>	2 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	60 Stunden

**Lernergebnisse:**

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

<b>Lehr- und Lernform:</b>	Projekt
<b>Leistungskontrolle:</b>	Präsentation
<b>Anteil Semesterwochenstunden:</b>	2 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	60 Stunden

**Lernergebnisse:**

Die Studierenden können Softwarearchitekturen von serviceorientierten Web-Anwendungen modellieren und die Technologien und Werkzeuge für serviceorientierten Web-Anwendungen in Projekten anwenden.

**Bildung der Modulnote:**

Klausur, unbenotetes Testat

## Modulbeschreibung Mobile Applikationen

**Schlüsselworte:** Apps, Client, Server, Soziale Netze

<b>Zielgruppe:</b>	1. oder 2. Semester AIM	<b>Modulnummer:</b>	AI W13
<b>Arbeitsaufwand:</b>	5 ECTS		150 h
<b>Davon</b>	<b>Kontaktzeit</b>		75 h
	<b>Selbststudium</b>		45 h
	<b>Prüfungsvorbereitung</b>		30 h
<b>Unterrichtssprache:</b>	Deutsch oder Englisch		
<b>Modulverantwortung:</b>	Prof. Dr.-Ing. Harald Melcher		
<b>Stand:</b>	11.12.2014		

### Voraussetzungen:

Kenntnisse in der objektorientierten Programmierung

### Gesamtziel:

Die Studierenden sind in der Lage Applikationen für ein gängiges Betriebssystem für mobile Endgeräte erstellen zu können.

### Inhalt:

- Plattformen und Programmiersprachen für mobile Applikationen
- Oberflächen-Entwurf und –Umsetzung
- Cross-Plattform-Aspekte
- Wirtschaftlichkeitsanalysen

### Literaturhinweise:

Apple Developer Network: [developer.apple.com](http://developer.apple.com)  
Android Developer Network: [developer.android.com](http://developer.android.com)

### Wird angeboten:

Einmal pro Jahr (Sommer- oder Wintersemester)

### Teilgebiete und Leistungsnachweise:

<b>Lehr- und Lernform:</b>	Vorlesung/Projekt
<b>Leistungskontrolle:</b>	Klausur (90 Minuten)
<b>Anteil Semesterwochenstunden:</b>	4 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	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, unbenotetes Testat



## Modulbeschreibung Mobile UX Design

**Schlüsselworte:** UX, UCD, Apps, Soziale Plattformen

<b>Zielgruppe:</b>	<b>1. oder 2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI W15</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>60 h</b>
	<b>Selbststudium</b>		<b>60 h</b>
	<b>Prüfungsvorbereitung</b>		<b>30 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch oder Englisch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Astrid Beck</b>		
<b>Stand:</b>	<b>01.07.2015</b>		

### Voraussetzungen:

Kenntnisse in der Programmierung

### Gesamtziel:

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.

Die Studierenden sind in der Lage Anforderungen, Konzeption und Design für mobile Endgeräte erstellen zu können, wie z.B. ein Online-Spiel, eine Marketing-Plattform oder eine App im Automotive-Umfeld.

Die Studierenden können die unterschiedlichen Realisierungsmöglichkeiten unterscheiden und bewerten: Web, mobile Webseiten, Responsive Design, Native und Hybride Apps.

Die Studierenden kennen das User Centered Design und insbesondere die Methoden User Research, Anforderungsermittlung, Prototyping sowie Usability Test für mobile Applikationen und können diese anwenden. Sie lernen Tools und Templates kennen und in den Phasen des User Centered Design einzusetzen.

Die Studierenden können Anforderungen des User Experience (UX) in Designs umsetzen. Die Studierenden können soziale Plattformen in Apps einbinden.

### Inhalt:

- Statistiken: Systeme, Mobile Nutzer, Anwendungen
- Mobile Usability, Mobile UX
- User Centered design
- Projektplanung
- Marktanalyse Wettbewerb
- 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:**

Jedes Semester

**Teilgebiete und Leistungsnachweise:**

<b>Lehr- und Lernform:</b>	Vorlesung und Projekt
<b>Leistungskontrolle:</b>	Gruppenpräsentation
<b>Anteil Semesterwochenstunden:</b>	4 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	150 Stunden

**Lernergebnisse:**

Die Studierenden sind in der Lage Anforderungen, Konzeption und Design für mobile Endgeräte erstellen zu können, wie z.B. ein Online-Spiel, eine Marketing-Plattform oder eine App im Automotive-Umfeld.

Die Studierenden können die unterschiedlichen Realisierungsmöglichkeiten unterscheiden und bewerten: Web, mobile Webseiten, Responsive Design, Native und Hybride Apps.

Die Studierenden kennen das User Centered Design und insbesondere die Methoden User Research, Anforderungsermittlung, Prototyping sowie Usability Test für mobile Applikationen und können diese anwenden. Sie lernen Tools und Templates kennen und in den Phasen des User Centered Design einzusetzen.

Die Studierenden können Anforderungen des User Experience (UX) in Designs umsetzen.  
Die Studierenden können soziale Plattformen in Apps einbinden..

**Bildung der Modulnote:**

Gruppenpräsentation und Projektdokumentation

## Modulbeschreibung Web Technologien

**Schlüsselworte:** Web-Anwendungen, Java Server Faces

<b>Zielgruppe:</b>	<b>1. oder 2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI W16</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>60 h</b>
	<b>Selbststudium</b>		<b>60 h</b>
	<b>Prüfungsvorbereitung</b>		<b>30 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch oder Englisch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Astrid Beck</b>		
<b>Stand:</b>	<b>01.07.2015</b>		

### Voraussetzungen:

Programmierkenntnisse in Java, Mobile UX Design

### Gesamtziel:

Ziel dieser Veranstaltung ist die Einführung in die Webentwicklung mit Hilfe des UI-Frameworks JSF (JavaServer Faces). JSF gehört zu den Webtechnologien der Java Enterprise Edition (Java EE).

Es wird auf die Grundlagen der Webentwicklung eingegangen und konkret mit Hilfe von JSF eine eigene Webanwendung realisiert.

Dabei werden die wichtigsten Konzepte in JSF behandelt, wie z.B. Bean Management, Navigation, Datenvalidierung oder Eventing.

Als Ausblick wird auf die Verwendung einer UI-Komponentenbibliothek eingegangen, wobei auch auf Ajax als Web 2.0-Technologie eingegangen wird.

In der Vorlesung werden sich theoretische und praktische Module abwechseln.

### Inhalt:

- Grundlagen Webentwicklung
- Entwicklung eines Webshops
- Phasen des UCD
- Konzeption, Planung, Steuerung und Einbindung von Social Media
- Vorstellung JavaServer Faces (JSF)
- Installation von JSF
- Ein erstes kleines Beispiel
- Navigationskonzept
- Datenkonvertierung und -validierung
- Eventing
- Model-View-Controller Design Pattern
- Integration von RichFaces in eine JSF-Anwendung
- Ajax-Grundlagen
- Verwendung verschiedener leistungsfähiger UI-Komponenten
- Bootstrap, BootsFaces

### Literaturhinweise:

Andy Bosch: Portlets und JavaServer Faces Broschiert, entwickler.Press 2009

Kai Laborenz und Andrea Ertel: Responsive Webdesign: Anpassungsfähige Websites programmieren und gestalten (Galileo Computing), 2014  
ISO 9241-11

### Wird angeboten:

Einmal pro Jahr (Sommer- oder Wintersemester)

**Teilgebiete und Leistungsnachweise:**

<b>Lehr- und Lernform:</b>	Vorlesung und Projekt
<b>Leistungskontrolle:</b>	Gruppenpräsentation
<b>Anteil Semesterwochenstunden:</b>	4 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	150 Stunden

**Lernergebnisse:**

Die Studierenden sind in der Lage Anforderungen, Konzeption und Design für einen Webshop erstellen zu können,.

Die Studierenden können nutzerzentriertes Design (UCD) anwenden und praktisch umsetzen

Die Studierenden können auf Basis der Ergebnisse des UserResearch, der Anforderungsermittlung, des Prototyping in eine lauffähige Webanwendung unter Zuhilfenahme von JavaServer Faces umsetzen.

Die Studierenden können Anforderungen des User Experience (UX) in Designs umsetzen.  
Die Studierenden können soziale Plattformen in Apps einbinden.

**Bildung der Modulnote:**

Gruppenpräsentation und Projektdokumentation

# Wahlpflichtmodule

## Vertiefung **Autonome Systeme**

### Inhaltsverzeichnis

<b>Modulnummer</b>	<b>Modul</b>	<b>Seite</b>
AI W03	<b>Mobile Communication</b>	<b>38</b>
AI W04	<b>Datenfusion</b>	<b>40</b>
AI W05	<b>Automotive System Design</b>	<b>41</b>
AI W06	<b>Advanced Control</b>	<b>42</b>
AI W07	<b>Systemtheorie</b>	<b>43</b>
AI W08	<b>Programmiermethoden der KI</b>	<b>45</b>
AI W09	<b>Self-Adaptive Software</b>	<b>47</b>
AI W10	<b>Software Tooling Workshop</b>	<b>49</b>

## Modulbeschreibung Mobile Communication

**Schlüsselworte:** Automotive, Communication, Safety, Security

<b>Zielgruppe:</b>	1. oder 2. Semester AIM	<b>Modulnummer:</b>	AI W03
<b>Arbeitsaufwand:</b>	5 ECTS		150 h
<b>Davon</b>	<b>Kontaktzeit</b>		50 h
	<b>Selbststudium</b>		100 h
	<b>Prüfungsvorbereitung</b>		0 h
<b>Unterrichtssprache:</b>	Englisch		
<b>Modulverantwortung:</b>	Prof. Dr.-Ing. Harald Melcher, Prof. Dr. Dominik Schoop, Prof. Dr.-Ing. Martin Zieher		
<b>Stand:</b>	20.12.2014		

### Voraussetzungen:

Grundlagen Kommunikationstechnik  
Module IT-Security Engineering and Advanced Software Engineering

### Gesamtziel:

Die Studierenden sind in der Lage:

- die Ziele von ITS und VANETs zu erklären
- die Architektur und Technologie von VANETs zu erklären
- die Sicherheit von VANETs einzuschätzen und Sicherheitsmaßnahmen vorzuschlagen
- einfache Car2Infrastructure- und Car2-Car-Applikationen zu implementieren

### Inhalt:

- Überblick Intelligent Transportation Systems (ITS)
- 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, ...)
- Zusammenhang Safety und Security
- Angriffe in VANETs
- Authentizität, Anonymität, Integrität, Vertraulichkeit und Missbrauchserkennung in VANETs
- pseudonyme PKIs

### Literaturhinweise:

Veröffentlichungen des Car-2-Car Communication Consortium (C2CCC), [www.car-2-car.org](http://www.car-2-car.org)  
Veröffentlichungen des Projektes „Sichere Intelligente Mobilität Testfeld Deutschland (simTD)“, [www.simtd.de](http://www.simtd.de)  
Veröffentlichungen des Projektes „Secure Vehicular Communication (SeVeCom)“, [www.sevecom.org](http://www.sevecom.org)  
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:

Einmal pro Jahr (Sommer- oder Wintersemester)

**Teilgebiete und Leistungsnachweise:**

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

**Lernergebnisse:**

Die Studierenden sind in der Lage:

- die Ziele von ITS und VANETs zu erklären
- die Architektur und Technologie von VANETs zu erklären

<b>Lehr- und Lernform:</b>	Projektarbeit (Labor)
<b>Leistungskontrolle:</b>	Abschlusspräsentation
<b>Anteil Semesterwochenstunden:</b>	0,5 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	90 Stunden

**Lernergebnisse:**

Die Studierenden sind in der Lage:

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

**Bildung der Modulnote:**

Klausur, unbenotetes Testat und unbenoteter Bericht

## Modulbeschreibung Datenfusion

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

<b>Zielgruppe:</b>	<b>1. oder 2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI W04</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>45 h</b>
	<b>Selbststudium</b>		<b>75 h</b>
	<b>Prüfungsvorbereitung</b>		<b>30 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Dr.-Ing. Reiner Marchthaler</b>		
<b>Stand:</b>	<b>10.12.2014</b>		

**Voraussetzungen:**

Grundlagen Statistik und Automatentheorie (im Speziellen in Zustandsautomaten)

**Gesamtziel:**

Die Studierenden sind in der Lage:

- Herleitung der Kalman-Filter-Gleichungen nachvollziehen können. Die hierzu zu Grunde liegenden Annahmen kennen und in der Lage sein Kalman-Filter zu entwerfen
- Problem aus der Technik in Form von Markov-Modell und Bayes-Netzte zu modellieren
- In praktischen Übungen Kalman-Filter in einem Mikrocontroller zu implementieren.

**Inhalt:**

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

**Literaturhinweise:**

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

Ole 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. New York, Chichester, Brisbane, Toronto, Singapore, Weinheim : John Wiley & Sons, Inc., 2012

**Wird angeboten:**

Einmal pro Jahr (Sommer- oder Wintersemester)

**Teilgebiete und Leistungsnachweise:**

**Lehr- und Lernform:** Vorlesung mit Übungen

**Leistungskontrolle:** Klausur (90 Minuten)

**Anteil Semesterwochenstunden:** 3 SWS

**Geschätzte studentische Arbeitszeit:** 120 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, Netzwerke und Kommunikationsprotokolle in der automobilen Anwendung, Funktionale Sicherheit

<b>Zielgruppe:</b>	1. oder 2. Semester AIM	<b>Modulnummer:</b>	AI W05
<b>Arbeitsaufwand:</b>	5 ECTS		150 h
<b>Davon</b>	<b>Kontaktzeit</b>		60 h
	<b>Selbststudium</b>		60 h
	<b>Prüfungsvorbereitung</b>		30 h
<b>Unterrichtssprache:</b>	Englisch		
<b>Modulverantwortung:</b>	Prof. Dr.-Ing. W. Zimmermann		
<b>Stand:</b>	09.07.2015		

### Voraussetzungen:

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

### Inhalt:

- Anwendungsdomänen (Powertrain, Chassis, Body, Infotainment, Driver Assistance)
- Grundlagen verteilter Systeme, Kommunikationsanforderungen, Analyse und Gestaltung unter Echtzeit- und Sicherheitsanforderungen
- Analyse der Zuverlässigkeit und der Sicherheit; funktionale Sicherheitsanforderungen und deren Implementierung

### 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 mit Gruppenprojekt und Präsentation  
**Leistungskontrolle:** Gruppenpräsentation, Klausur (60 min)  
**Anteil Semesterwochenstunden:** 4 SWS  
**Geschätzte studentische Arbeitszeit:** 150 Stunden

### Lernergebnisse:

- Die Studierenden
- verstehen die Architektur von Systemen im Automobil und deren Software
  - sind in der Lage, die Netzwerkkommunikation dieser Systeme zu gestalten und quantitativ abzuschätzen
  - können die Systeme hinsichtlich Sicherheit und Zuverlässigkeit abschätzen.

### Bildung der Modulnote:

Klausur, unbenotetes Testat

## Modulbeschreibung Advanced Control

**Schlüsselworte:** Modellierung und Simulation von Regelschleifen mit MATLAB/SIMULINK; moderne Regelungstechnik (Zustandsregler und Zustandsbeobachter)

<b>Zielgruppe:</b>	1. oder 2. Semester AIM	<b>Modulnummer:</b>	AI W06
<b>Arbeitsaufwand:</b>	5 ECTS		150 h
<b>Davon</b>	<b>Kontaktzeit</b>		60 h
	<b>Selbststudium</b>		60 h
	<b>Prüfungsvorbereitung</b>		30 h
<b>Unterrichtssprache:</b>	Englisch oder Deutsch		
<b>Modulverantwortung:</b>	Prof. Dr. Walter Lindermeir		
<b>Stand:</b>	30.07.2015		

### Voraussetzungen:

Kenntnisse einer Simulationssprache (z.B.. MATLAB/Simulink)  
Grundlegende Kenntnisse der Steuerungs- und Regelungstechnik (Transferfunktionen, Laplace)

### Inhalt:

- Zustandsbeschreibungen von dynamischen Systemen und die Beziehung zu Laplace Transformationen
- Entwicklung von Regelsystemen im Zustandsraum
- Entwicklung von Beobachtern
- Entwicklung von Digitalreglern und diskreten Zustandsräumen

### Literaturhinweise:

Ogata, J.: Modern Control Engineering  
Föllinger, O.: Regelungstechnik. Hüthig Verlag  
Lunze, J.: Regelungstechnik. 2 Bände, Springer Verlag  
Unbehauen, H.: Regelungstechnik. 3 Bände, Vieweg Verlag

### Wird angeboten:

Wintersemester

### Teilgebiete und Leistungsnachweise:

**Lehr- und Lernform:** Vorlesung  
**Leistungskontrolle:** Klausur (35 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 Zustandsregler und Zustandsbeobachter für zu regelnde Strecken eigenständig auslegen. Dies gilt sowohl für die analytische als auch für die numerische Auslegung mit dem Werkzeug Matlab/Simulink.

### Bildung der Modulnote:

Klausur

## Modulbeschreibung Systemtheorie für Informatiker

**Schlüsselworte:** Mathematische Beschreibung technischer Systeme

<b>Zielgruppe:</b>	<b>1. oder 2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI W07</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>60 h</b>
	<b>Selbststudium</b>		<b>60 h</b>
	<b>Prüfungsvorbereitung</b>		<b>30 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Dr.-Ing. Reinhard Schmidt</b>		
<b>Stand:</b>	<b>25.12.2014</b>		

### Voraussetzungen:

Mathematik aus Bachelor-Studium

### Gesamtziel:

Die Studierenden erwerben die Kompetenz technische Systeme auf mathematische Weise zu beschreiben, zu analysieren und zu verstehen. Sie sind in der Lage, mathematische Problemlösungsmöglichkeiten für technische System im Bereich der Forschung anzuwenden.

Ziel dieses Moduls:

Die Studierenden beherrschen die mathematischen Kenntnisse analoge, zeitdiskrete, nichtlineare und chaotische Systeme zu beschreiben und zu analysieren. Sie beherrschen die mathematischen Methoden deterministische und stochastische Signale zu beschreiben. Sie beherrschen Methoden zur Signaldetektion und Signalschätzung.

### Inhalt:

Beschreibung technischer Systeme

- Zeitkontinuierliche Systeme
- Ein- und mehrdimensionale zeitdiskrete Systeme
- Zeitdiskrete Filterbänke
- Adaptive Systeme
- Chaotische Systeme
- Nichtlineare Systeme

Beschreibung von Signalen

- Deterministische Signale
- Stochastische Signale

Transformationen der Systemtheorie

- Fourier-Transformation
- Laplace-Transformation
- z-Transformation
- Wavelet-Transformation
- Leistungsdichtespektrum, Wiener-Chintschin-Theorem,
- Karhunen-Loeve-Transformation

Zahlentheoretische Methoden in der Signalverarbeitung

- Faltung und schnelle Fouriertransformation

Methoden der Signaldetektion

Methoden der Signalschätzung

- Filterung
- Prädiktion
- Interpolation

**Literaturhinweise:**

Rolf Unbehauen: Systemtheorie Band 1+2, Oldenbourg Verlag.  
Kristian Kroschel: Statische Nachrichtentheorie Band 1+2, Springer Verlag.  
Bernd Girod, Rudolf Rabenstein, Alexander Stenger: Einführung in die Systemtheorie,  
Teubner Verlag

**Wird angeboten:**

Einmal pro Jahr (Sommer- oder Wintersemester)

**Teilgebiete und Leistungsnachweise:**

**Lehr- und Lernform:** Vorlesung mit Übungen, Nachbereitung und  
Prüfungsvorbereitung

**Leistungskontrolle:** Klausur (90 Minuten)

**Anteil Semesterwochenstunden:** 4 SWS

**Geschätzte studentische Arbeitszeit:** 120 Stunden

**Lernergebnisse:**

Die Studierenden beherrschen die mathematischen Konzepte der Systemtheorie. Sie können technische Systeme mathematisch beschreiben und selbstständig Problemlösungen ableiten und weiterentwickeln.

**Bildung der Modulnote:**

Klausur

## Modulbeschreibung Programmiermethoden der KI

**Schlüsselworte:** Theoretische Informatik

<b>Zielgruppe:</b>	<b>1. oder 2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI W08</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>60 h</b>
	<b>Selbststudium</b>		<b>60 h</b>
	<b>Prüfungsvorbereitung</b>		<b>30 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Dr. Heinrich Weber</b>		
<b>Stand:</b>	<b>02.01.2015</b>		

**Voraussetzungen:**

Eine höhere imperative Programmiersprache, Suchalgorithmen sowie elementare Logik

**Gesamtziel:**

Die Studierenden lernen Probleme und ihre Lösungen als ausgezeichnete Zustände in einem Zustandsraum zu betrachten. Die Lösungsfindung gestaltet sich hier als gerichtete Transformation von Zuständen vom Ausgangszustand zu einer oder den möglichen Lösungen. Sie lernen dabei den Umgang mit der KI-Sprache PROLOG sowie die der Sprache zu Grunde liegenden Konzepte und Logikkalküle. Die Kenntnisse in der logischen Programmierung werden in praktischen Übungen zur Lösung typischer KI-Probleme gefestigt.

**Inhalt:**

- Ableitungssysteme
- nicht rücksetzende und rücksetzende Suchstrategien
- Navigation in Zustandsräumen, Breitensuche und Tiefensuche
- Algorithmenkonstruktion mit Hilfe von Ableitungssystemen
- Prädikatenlogik
- Normalformen
- Klauselmengen , Horn-Klauseln
- Termreduktionssysteme
- Unifikations-Algorithmus
- Resolutionskalkül
- Einführung in PROLOG
- klassische Beispiele und Übungsaufgaben zur logischen Programmierung

**Literaturhinweise:**

Clocksinn, W. F.: Programming in Prolog, Springer, 2013.  
Schöning, U.: Logik für Informatiker, Spektrum Akademischer Verlag, 2000.

**Wird angeboten:**

Wintersemester

**Teilgebiete und Leistungsnachweise:**

<b>Lehr- und Lernform:</b>	Vorlesung
<b>Leistungskontrolle:</b>	Klausur (90 Minuten)
<b>Anteil Semesterwochenstunden:</b>	3 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	90 Stunden

**Lernergebnisse:**

Die Studierenden sind in der Lage, Probleme und ihre Lösungen als ausgezeichnete Zustände in einem Zustandsraum zu betrachten. Die Lösungsfindung gestaltet sich hier als gerichtete Transformation von Zuständen vom Ausgangszustand zu einer oder den möglichen Lösungen. Sie beherrschen den Umgang mit der KI-Sprache PROLOG sowie die der Sprache zu Grunde liegenden Konzepte und Logikkalküle.

<b>Lehr- und Lernform:</b>	Übungen (Labor)
<b>Leistungskontrolle:</b>	Klausur
<b>Anteil Semesterwochenstunden:</b>	1 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	60 Stunden

**Lernergebnisse:**

Die Studierenden beherrschen den praktischen Umgang mit einer KI Sprache.

**Bildung der Modulnote:**

Klausur, unbenotetes Testat

## Modulbeschreibung Self-Adaptive Software

### Schlüsselworte: Software Self-\* Techniques

<b>Zielgruppe:</b>	<b>1. oder 2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI W09</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>30 h</b>
	<b>Selbststudium</b>		<b>120 h</b>
	<b>Prüfungsvorbereitung</b>		<b>0 h</b>
<b>Unterrichtssprache:</b>	<b>Englisch oder Deutsch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Hans-Gerhard Groß</b>		
<b>Stand:</b>	<b>16.12.2014</b>		

### Voraussetzungen:

Grundlagen des Softwareengineering, Software Architektur, Distributed Systems

### Gesamtziel:

Die Studierenden sind in der Lage:

- Geeignete Techniken zur Realisierung von adaptiven Systemen zu evaluieren, auszuwählen und integrieren.
- System Monitoring zu entwickeln und effizient einzusetzen.
- Diagnoseverfahren zu entwickeln und effizient einzusetzen.
- Neue Methoden und Techniken zu entwickeln, die eine automatische Software Adaption ermöglichen.

### Inhalt:

- Individuelle Projektarbeit zum Thema Software Self-\* Techniques.
- Adaptive Software Platforms, Software Monitoring and Diagnosis, Software Healing
- 24-7 Uptime Requirements of Software Systems.
- Entwicklung eines Prototypen, der die gewählte Self-\* Technique realisiert.

### Literaturhinweise:

Cheng, B.H.C., de Lemos, R., Inverardi, P., Magee, J. (Eds.) Software Engineering for Self-Adaptive Systems, Springer LNCS 5525, 2009.

Individuelle Literatursuche entspr. des gewählten Themas (Literature Search).

### Wird angeboten:

Einmal pro Jahr (Sommer- oder Wintersemester)

### Teilgebiete und Leistungsnachweise:

<b>Lehr- und Lernform:</b>	Seminar
<b>Leistungskontrolle:</b>	Seminararbeit
<b>Anteil Semesterwochenstunden:</b>	1 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	30 Stunden

### Lernergebnisse:

Die Studierenden sind in der Lage aktuelle Themen im Bereich Software Self-\* Techniques selbstständig zu recherchieren und zu präsentieren.

**Lehr- und Lernform:** Projekt  
**Leistungskontrolle:** Projektarbeit  
**Anteil Semesterwochenstunden:** 2 SWS  
**Geschätzte studentische Arbeitszeit:** 60 Stunden

**Lernergebnisse:**

Die Studierenden sind in der Lage:

- Geeignete Techniken zur Realisierung von adaptiven Systemen zu evaluieren, auszuwählen und integrieren.
- System Monitoring zu entwickeln und effizient einzusetzen.
- Diagnoseverfahren zu entwickeln und effizient einzusetzen.
- Neue Methoden und Techniken zu entwickeln, die eine automatische Software Adaption ermöglichen.

**Lehr- und Lernform:** Online-Kurs, Selbststudium  
**Leistungskontrolle:** Hausarbeit  
**Anteil Semesterwochenstunden:** 1 SWS  
**Geschätzte studentische Arbeitszeit:** 30 Stunden

**Lernergebnisse:**

Die Studierenden besitzen Kenntnisse über die Methoden von adaptiven Software Systemen.

**Bildung der Modulnote:**

Klausur



## Modulbeschreibung Software Tooling Workshop

**Schlüsselworte:** Software Tools, Qualität, DSL

<b>Zielgruppe:</b>	<b>1. oder 2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI W10</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>30 h</b>
	<b>Selbststudium</b>		<b>120 h</b>
	<b>Prüfungsvorbereitung</b>		<b>0 h</b>
<b>Unterrichtssprache:</b>	<b>Englisch oder Deutsch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Hans-Gerhard Groß</b>		
<b>Stand:</b>	<b>16.12.2014</b>		

**Voraussetzungen:**

Grundlagen des Softwareengineering, Software Architektur

**Gesamtziel:**

Die Studierenden sind in der Lage:

- Existierende Theorien des Software Quality Management in entsprechende Tools zur Messung von Qualität umzusetzen und prototypisch zu realisieren.
- Tools für ausgewählte Phasen des Software-Lebenszyklus zu entwickeln und zu evaluieren.
- Moderne Sprachen der 4. Generation effizient für die schnelle Entwicklung von Software-Tools anzuwenden.

**Inhalt:**

- Individuelle Projektarbeit zum Thema Software Tools.
- Entwicklung eines Tool-Prototyps.
- Evaluierung der Technologien plus Evaluierung des Tools mittels Fallstudie.
- Dokumentation der zugrunde liegenden Technologien, des Tools, und der Evaluierung

**Literaturhinweise:**

Individuelle Literatursuche entspr. des gewählten Themas (Literature Search).

**Wird angeboten:**

Einmal pro Jahr (Sommer- oder Wintersemester)

**Teilgebiete und Leistungsnachweise:**

<b>Lehr- und Lernform:</b>	Seminar
<b>Leistungskontrolle:</b>	Seminararbeit
<b>Anteil Semesterwochenstunden:</b>	1 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	30 Stunden

**Lernergebnisse:**

Die Studierenden sind in der Lage aktuelle Themen im Bereich Software Tooling selbständig zu recherchieren und zu präsentieren.

<b>Lehr- und Lernform:</b>	Projekt
<b>Leistungskontrolle:</b>	Projektarbeit
<b>Anteil Semesterwochenstunden:</b>	3 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	90 Stunden

**Lernergebnisse:**

Die Studierenden sind in der Lage:

- Existierende Theorien des Software Quality Management in entsprechende Tools zur Messung von Qualität umzusetzen und prototypisch zu realisieren.
- Tools für ausgewählte Phasen des Software-Lebenszyklus zu entwickeln und zu evaluieren.
- Moderne Sprachen der 4. Generation effizient für die schnelle Entwicklung von Software-Tools anzuwenden

**Bildung der Modulnote:**

Bericht und Referat benotet, unbenotetes Testat

# Wahlpflichtmodule

## Vertiefung **Business Intelligence**

### Inhaltsverzeichnis

<b>Modulnummer</b>	<b>Modul</b>	<b>Seite</b>
AI W01	<b>Intelligent Data Analytics</b>	<b>52</b>
AI W02	<b>Business Intelligence</b>	<b>54</b>
AI W03	<b>Datenfusion</b>	<b>56</b>
AI W04	<b>Programmiermethoden der KI</b>	<b>57</b>
AI W08	<b>Rechnerorganisation</b>	<b>59</b>
AI W11	<b>Web Services</b>	<b>61</b>
AI W12	<b>Mobile Applikationen</b>	<b>63</b>
AI W13	<b>Data Mining</b>	<b>64</b>

## Modulbeschreibung Intelligent Data Analytics

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

<b>Zielgruppe:</b>	1. oder 2. Semester AIM	<b>Modulnummer:</b>	AI W01
<b>Arbeitsaufwand:</b>	5 ECTS		150 h
<b>Davon</b>	<b>Kontaktzeit</b>		30 h
	<b>Selbststudium</b>		90 h
	<b>Prüfungsvorbereitung</b>		30 h
<b>Unterrichtssprache:</b>	Englisch		
<b>Modulverantwortung:</b>	Prof. Jürgen Nonnast		
<b>Stand:</b>	07.07.2015		

### Voraussetzungen:

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

### Gesamtziel:

Der Student lernt Anwendungen kennen, in denen Zeitreihen generiert und aufgezeichnet werden. Beispielhaft seien die Automobilindustrie, der Finanzsektor und Daten aus dem Internet oder den sozialen Medien genannt. In diesem Modul erlernt der Student Algorithmen aus den Bereichen „Data Mining“ und „maschinelles Lernen“, und erwirbt die Kompetenz, Zeitreihen intelligent zu 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:

Einmal pro Jahr (Sommer- oder Wintersemester)

**Teilgebiete und Leistungsnachweise:**

<b>Lehr- und Lernform:</b>	Vorlesung mit praktischen Übungen
<b>Leistungskontrolle:</b>	Individuelle Seminararbeit und Seminarvortrag
<b>Anteil Semesterwochenstunden:</b>	4 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	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 Modell-basierten 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:**

Seminararbeit und Seminarvortrag

## Modulbeschreibung Business Intelligence

**Schlüsselworte:** Business Intelligence, Datawarehouse, OLAP, Business Performance

<b>Zielgruppe:</b>	<b>1. oder 2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI W02</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>30 h</b>
	<b>Selbststudium</b>		<b>90 h</b>
	<b>Prüfungsvorbereitung</b>		<b>30 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Dr. Dirk Hesse</b>		
<b>Stand:</b>	<b>23.12.2014</b>		

### Voraussetzungen:

Grundlagen Datenbanken

### Gesamtziel:

Die Studierenden beherrschen die Grundbegriffe der Business Intelligence und kennen die Bedeutung für die unternehmerische Praxis. Sie verfügen über Kenntnisse der verschiedenen Konzepte und Methoden der Business Intelligence. Sie besitzen die Fertigkeit verschiedene Instrumente und Anwendungen der Business Intelligence zielgerichtet einzusetzen und haben die Kompetenz für spezifische Anwendungsfelder geeignete Verfahren auszuwählen und umzusetzen.

### 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:

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:

Einmal pro Jahr (Sommer- oder Wintersemester)

### Teilgebiete und Leistungsnachweise:

<b>Lehr- und Lernform:</b>	Vorlesung mit Seminar
<b>Leistungskontrolle:</b>	Individuelle Seminararbeit und Seminarvortrag
<b>Anteil Semesterwochenstunden:</b>	1,5 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	60 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.

<b>Lehr- und Lernform:</b>	Projektarbeit (Labor)
<b>Leistungskontrolle:</b>	Abschlusspräsentation
<b>Anteil Semesterwochenstunden:</b>	0,5 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	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:**

Modulnote: Klausur 90 Minuten,  
Laborteil: Testat, unbenotet

## Modulbeschreibung Datenfusion

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

<b>Zielgruppe:</b>	<b>1. oder 2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI W04</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>45 h</b>
	<b>Selbststudium</b>		<b>75 h</b>
	<b>Prüfungsvorbereitung</b>		<b>30 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Dr.-Ing. Reiner Marchthaler</b>		
<b>Stand:</b>	<b>10.12.2014</b>		

### Voraussetzungen:

Grundlagen Statistik und Automatentheorie (im Speziellen in Zustandsautomaten)

### Gesamtziel:

Die Studierenden sind in der Lage:

- Herleitung der Kalman-Filter-Gleichungen nachvollziehen können. Die hierzu zu Grunde liegenden Annahmen kennen und in der Lage sein Kalman-Filter zu entwerfen
- Problem aus der Technik in Form von Markov-Modell und Bayes-Netzte zu modellieren
- In praktischen Übungen Kalman-Filter in einem Mikrocontroller zu implementieren.

### Inhalt:

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

### Literaturhinweise:

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

Ole 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. New York, Chichester, Brisbane, Toronto, Singapore, Weinheim : John Wiley & Sons, Inc., 2012

### Wird angeboten:

Einmal pro Jahr (Sommer- oder Wintersemester)

### Teilgebiete und Leistungsnachweise:

**Lehr- und Lernform:** Vorlesung mit Übungen

**Leistungskontrolle:** Klausur (90 Minuten)

**Anteil Semesterwochenstunden:** 3 SWS

**Geschätzte studentische Arbeitszeit:** 120 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 Programmiermethoden der KI

**Schlüsselworte:** Theoretische Informatik

<b>Zielgruppe:</b>	1. oder 2. Semester AIM	<b>Modulnummer:</b>	AI W08
<b>Arbeitsaufwand:</b>	5 ECTS		150 h
<b>Davon</b>	<b>Kontaktzeit</b>		60 h
	<b>Selbststudium</b>		60 h
	<b>Prüfungsvorbereitung</b>		30 h
<b>Unterrichtssprache:</b>	Deutsch		
<b>Modulverantwortung:</b>	Prof. Dr. Heinrich Weber		
<b>Stand:</b>	02.01.2015		

**Voraussetzungen:**

Eine höhere imperative Programmiersprache, Suchalgorithmen sowie elementare Logik

**Gesamtziel:**

Die Studierenden lernen Probleme und ihre Lösungen als ausgezeichnete Zustände in einem Zustandsraum zu betrachten. Die Lösungsfindung gestaltet sich hier als gerichtete Transformation von Zuständen vom Ausgangszustand zu einer oder den möglichen Lösungen. Sie lernen dabei den Umgang mit der KI-Sprache PROLOG sowie die der Sprache zu Grunde liegenden Konzepte und Logikkalküle. Die Kenntnisse in der logischen Programmierung werden in praktischen Übungen zur Lösung typischer KI-Probleme gefestigt.

**Inhalt:**

- Ableitungssysteme
- nicht rücksetzende und rücksetzende Suchstrategien
- Navigation in Zustandsräumen, Breitensuche und Tiefensuche
- Algorithmenkonstruktion mit Hilfe von Ableitungssystemen
- Prädikatenlogik
- Normalformen
- Klauselmengen , Horn-Klauseln
- Termreduktionssysteme
- Unifikations-Algorithmus
- Resolutionskalkül
- Einführung in PROLOG
- klassische Beispiele und Übungsaufgaben zur logischen Programmierung

**Literaturhinweise:**

Clocksinn, W. F.: Programming in Prolog, Springer, 2013.  
Schöning, U.: Logik für Informatiker, Spektrum Akademischer Verlag, 2000.

**Wird angeboten:**

Wintersemester

**Teilgebiete und Leistungsnachweise:**

<b>Lehr- und Lernform:</b>	Vorlesung
<b>Leistungskontrolle:</b>	Klausur (90 Minuten)
<b>Anteil Semesterwochenstunden:</b>	3 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	90 Stunden

**Lernergebnisse:**

Die Studierenden sind in der Lage, Probleme und ihre Lösungen als ausgezeichnete Zustände in einem Zustandsraum zu betrachten. Die Lösungsfindung gestaltet sich hier als gerichtete Transformation von Zuständen vom Ausgangszustand zu einer oder den möglichen Lösungen. Sie beherrschen den Umgang mit der KI-Sprache PROLOG sowie die der Sprache zu Grunde liegenden Konzepte und Logikkalküle.

<b>Lehr- und Lernform:</b>	Übungen (Labor)
<b>Leistungskontrolle:</b>	Klausur
<b>Anteil Semesterwochenstunden:</b>	1 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	60 Stunden

**Lernergebnisse:**

Die Studierenden beherrschen den praktischen Umgang mit einer KI Sprache.

**Bildung der Modulnote:**

Klausur, unbenotetes Testat

## Modulbeschreibung Rechnerorganisation

### Schlüsselworte: Praktische Informatik

<b>Zielgruppe:</b>	<b>1. oder 2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI W11</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>60 h</b>
	<b>Selbststudium</b>		<b>60 h</b>
	<b>Prüfungsvorbereitung</b>		<b>30 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Dr. Heinrich Weber</b>		
<b>Stand:</b>	<b>02.01.2015</b>		

### Voraussetzungen:

Vorlesung Betriebssysteme, eine Programmiersprache

### Gesamtziel:

Die Studierenden lernen die erweiterte Systemprogrammierung und Systemkonfiguration zur Erstellung und zum Betrieb eines verteilten, heterogenen Rechnernetzes, wie es in Rechenzentren oder von Dienstleistungs Providern vorgehalten wird. Dies umfasst die Integration von Hardwarekomponenten und Treibersoftware ebenso wie die Einrichtung und den Betrieb der benötigten Rechnerdienste (z.B. DHCP/BOOTP, BIND, NFS, NIS, LDAP, HTTPD, SAMBA, etc.) In die Integration der Dienste werden Verfahren zur Abwehr von Rechnerangriffen und Methoden der Computerforensik mit einbezogen.

### Inhalt:

- Rechner- und Betriebssystem-Bootvorgang
- Einbinden spezieller Rechner-Hardware, Treibermodule
- System-V und andere Upstart-Services
- Netzwerkintegration und einfache Netzwerkservices
- Erweiterte Netzwerkdienste
- Userverwaltung und Authentifizierung in großen Netzen
- Serverdienste für größere Netzwerke
- Windows, Unix und Mac in einem Netzwerk
- Sicherheitsaspekte und Abwehr von Angriffen

### Literaturhinweise:

Scott, M. Mitchell, E.L.: Linux System Security, Prentice Hall, 2-nd Ed., 2002  
Kettner, M.: Fehlerdiagnose und Problembhebung unter Linux, Suse Press, 2004

### Wird angeboten:

Wintersemester

**Teilgebiete und Leistungsnachweise:**

<b>Lehr- und Lernform:</b>	Vorlesung
<b>Leistungskontrolle:</b>	Klausur (90 Minuten)
<b>Anteil Semesterwochenstunden:</b>	3 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	90 Stunden

**Lernergebnisse:**

Die Studierenden kennen die erweiterte Systemprogrammierung und Systemkonfiguration zur Erstellung und zum Betrieb eines verteilten, heterogenen Rechnernetzes, wie es in Rechenzentren oder von Dienstleistungs Providern vorgehalten wird. Dies umfasst die Integration von Hardwarekomponenten und Treibersoftware ebenso wie die Einrichtung und den Betrieb der benötigten Rechnerdienste (z.B. DHCP/BOOTP, BIND, NFS, NIS, LDAP, HTTPD, SAMBA, etc.) In die Integration der Dienste werden Verfahren zur Abwehr von Rechnerangriffen und Methoden der Computerforensik mit einbezogen.

<b>Lehr- und Lernform:</b>	Übungen (Labor)
<b>Leistungskontrolle:</b>	Testat
<b>Anteil Semesterwochenstunden:</b>	1 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	60 Stunden

**Lernergebnisse:**

Die Studierenden sind in der Lage ein verteiltes, heterogenes Rechnernetz zu konfigurieren und in Betrieb zu nehmen.

**Bildung der Modulnote:**

Klausur, unbenotetes Testat

## Modulbeschreibung Web Services

**Schlüsselworte:** Serviceorientierte Architekturen, Web-Anwendungen

<b>Zielgruppe:</b>	<b>1. oder 2. Semester AIM</b>	<b>Modulnummer:</b>	<b>AI W12</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>60 h</b>
	<b>Selbststudium</b>		<b>60 h</b>
	<b>Prüfungsvorbereitung</b>		<b>30 h</b>
<b>Unterrichtssprache:</b>	<b>Deutsch oder Englisch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Dr.-Ing. Andreas Rößler</b>		
<b>Stand:</b>	<b>30.12.2014</b>		

**Voraussetzungen:**

Grundlegende Web-Technologien, Rechnernetze, Softwareentwicklung

**Gesamtziel:**

Die Studierenden können verteilte Systeme auf der Basis von Web Services aufbauen.

**Inhalt:**

- Grundlagen der Web Services
- Serviceorientierte Architekturen
- Verteilte Anwendungen
- Middleware
- SOAP, SOAP over HTTP
- Web Services Description Language
- UDDI: Universal Description, Discovery and Integration
- Sichere Web Services

**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:**

Einmal pro Jahr (Sommer- oder Wintersemester)

**Teilgebiete und Leistungsnachweise:**

<b>Lehr- und Lernform:</b>	Vorlesung / Seminar
<b>Leistungskontrolle:</b>	Präsentation (30 Minuten)
<b>Anteil Semesterwochenstunden:</b>	2 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	60 Stunden

**Lernergebnisse:**

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

<b>Lehr- und Lernform:</b>	Projekt
<b>Leistungskontrolle:</b>	Präsentation
<b>Anteil Semesterwochenstunden:</b>	2 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	60 Stunden

**Lernergebnisse:**

Die Studierenden können Softwarearchitekturen von serviceorientierten Web-Anwendungen modellieren und die Technologien und Werkzeuge für serviceorientierten Web-Anwendungen in Projekten anwenden.

**Bildung der Modulnote:**

Klausur, unbenotetes Testat

## Modulbeschreibung Mobile Applikationen

**Schlüsselworte:** Apps, Client, Server, Soziale Netze

<b>Zielgruppe:</b>	1. oder 2. Semester AIM	<b>Modulnummer:</b>	AI W13
<b>Arbeitsaufwand:</b>	5 ECTS		150 h
<b>Davon</b>	<b>Kontaktzeit</b>		75 h
	<b>Selbststudium</b>		45 h
	<b>Prüfungsvorbereitung</b>		30 h
<b>Unterrichtssprache:</b>	Deutsch oder Englisch		
<b>Modulverantwortung:</b>	Prof. Dr.-Ing. Harald Melcher		
<b>Stand:</b>	11.12.2014		

### Voraussetzungen:

Kenntnisse in der objektorientierten Programmierung

### Gesamtziel:

Die Studierenden sind in der Lage Applikationen für ein gängiges Betriebssystem für mobile Endgeräte erstellen zu können.

### Inhalt:

- Plattformen und Programmiersprachen für mobile Applikationen
- Oberflächen-Entwurf und –Umsetzung
- Cross-Plattform-Aspekte
- Wirtschaftlichkeitsanalysen

### Literaturhinweise:

Apple Developer Network: [developer.apple.com](http://developer.apple.com)  
Android Developer Network: [developer.android.com](http://developer.android.com)

### Wird angeboten:

Einmal pro Jahr (Sommer- oder Wintersemester)

### Teilgebiete und Leistungsnachweise:

<b>Lehr- und Lernform:</b>	Vorlesung/Projekt
<b>Leistungskontrolle:</b>	Klausur (90 Minuten)
<b>Anteil Semesterwochenstunden:</b>	4 SWS
<b>Geschätzte studentische Arbeitszeit:</b>	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, unbenotetes Testat

## Modulbeschreibung Data Mining

### Schlüsselworte: Mathematische Beschreibung technischer Systeme

<b>Zielgruppe:</b>	<b>1. oder 2.Semester AIM</b>	<b>Modulnummer:</b>	<b>AI W14</b>
<b>Arbeitsaufwand:</b>	<b>5 ECTS</b>		<b>150 h</b>
<b>Davon</b>	<b>Kontaktzeit</b>		<b>60 h</b>
	<b>Selbststudium</b>		<b>60 h</b>
	<b>Prüfungsvorbereitung</b>		<b>30 h</b>
<b>Unterrichtssprache:</b>	<b>Englisch</b>		
<b>Modulverantwortung:</b>	<b>Prof. Jürgen Nonnast</b>		
<b>Stand:</b>	<b>07.07.2015</b>		

#### Voraussetzungen:

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

#### Gesamtziel:

Die Studierenden erwerben die Kompetenz Information aus Daten zu gewinnen.

Ziele dieses Moduls:

Die Studierenden erwerben theoretische und praktische Kenntnisse über grundlegende Techniken des Data-Mining und deren Anwendungsgebiete.

#### Inhalt:

Visuelle Datenanalyse:

- Streudiagramme, Histogramme, (parallele) Boxplots, Barplots, Linienplots, parallele Koordinatenplots
- Interaktive Plots zur Entdeckung von Zusammenhängen

Unüberwachtes Lernen (Clustering):

- Distanzmaße
- Partitionierende Clusterverfahren
- Hierarchische Clusterverfahren

Überwachtes Lernen (Klassifikation):

- k-Nächste-Nachbarn-Klassifikation
- Neuronale Netze
- Support Vektor Maschinen
- Modell- und Merkmalsauswahl

#### Literaturhinweise:

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

Michael R. Berthold: Guide to intelligent data analysis: how to intelligently make sense of real data. - London [u.a.]: Springer, 2010. - ISBN: 9781848822597

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

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

Berthold, Borgelt, Höppner, Klawonn: Guide to Intelligent Data Analysis - How to Intelligently Make Sense of Real Data, Springer, 2012. - ISBN: 978-1447125723

#### Wird angeboten:

Einmal pro Jahr (Sommer- oder 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 Vorlesung besteht aus einem theoretischen und einem praktischen Teil, die sich gegenseitig abwechseln. Die Studierenden lernen grundlegende Methoden zur Datenvisualisierung kennen und wenden diese anhand von Open-Source-Softwareprodukten auf Datensätze an. Ferner sind Sie mit einigen gängigen Cluster- und Klassifikationsverfahren vertraut und können diese praktisch anwenden.

**Bildung der Modulnote:**

Klausur