



Modulhandbuch

Informatik - Master

Fakultät für Elektrotechnik und Informatik

Stand: 2018-07-09

Inhalt

1	Einführung und Studienaufbau	3
1.1	Studienziel und Kompetenzprofil	3
1.2	Studienabschluss.....	3
1.3	Studienaufbau.....	4
1.4	Qualifikationsvoraussetzungen.....	4
1.5	Fachstudienberatung.....	5
1.6	Studiengangleitung	5
2	Curriculare Struktur	6
2.1	Allgemeine Pflichtfächer	6
2.2	Studienschwerpunkte	8
3	Modulbeschreibungen	9
3.1	Allgemeine Pflichtmodule.....	9
	Architektur- und Entwurfsmuster der Softwaretechnik	9
	Komplexität von Algorithmen und deren Optimierung	11
	Software-Engineering für skalierbare Anwendungen	13
	Angewandte Logik für Modellierung und Verifikation.....	15
	Seminar zu Themen der Informatik	17
	Projekt.....	19
	Masterarbeit	21
	Seminar zur Stärkung der Schlüsselqualifikationen	23
3.2	Pflichtmodule Schwerpunkt "Security and Safety"	25
	Sicherheit moderner Netzwerke.....	25
	Security-Engineering in der IT	27
	Computer-Forensik	29
	Normen und Prozesse zur Entwicklung sicherheitskritischer Systeme	31
	Software-Technik für sicherheitskritische Systeme	33
3.3	Pflichtmodule Schwerpunkt "Information Systems Engineering"	35
	Daten-Management und -Analyse	35
	Hochleistungs-Datenhaltungs-Systeme	37
	Enterprise-Architecture-Management	39
	IT-Integrations- und Migrationstechnologien	41
	Implementierung von Informationssystemen	43

1 Einführung und Studienaufbau

1.1 Studienziel und Kompetenzprofil

Der weiterqualifizierende Master-Studiengang Informatik baut inhaltlich auf dem grundständigen Bachelor-Studiengang Informatik der Technischen Hochschule Ingolstadt auf. Bewerber verfügen bereits über ein solides Grundlagenwissen, Methodenkompetenz und eine Reihe von überfachlichen Qualifikationen.

Ziel des Master-Studiengangs ist es, zum einen die theoretisch-wissenschaftlichen Grundlagen der Studierenden zu verbreitern, um ihnen wahlweise eine Promotion bzw. die Arbeit im wissenschaftlichen Bereich zu ermöglichen und zum anderen - als stärker anwendungsorientierter Master-Studiengang - den Studierenden eine wesentliche Vertiefung in einem speziellen Anwendungsgebiet zu vermitteln. Die Vertiefungsgebiete orientieren sich am Profil der Technischen Hochschule Ingolstadt.

Darüber hinaus werden die analytische Kompetenz, die Methodenkompetenz und die Schlüsselqualifikationen der Studierenden weiter gestärkt, ihre Fähigkeit zur Reflexion des eigenen Handelns und Verhaltens geschult und Führungswissen und Führungstechniken vermittelt.

Der Master qualifiziert wahlweise für eine spätere Tätigkeit im wissenschaftlichen Bereich (dies schließt die Möglichkeit der Promotion ein), im höheren Dienst bei öffentlichen Einrichtungen oder für eine Position als Projektleiter oder als Führungskraft in Unternehmen mit internationaler Ausrichtung.

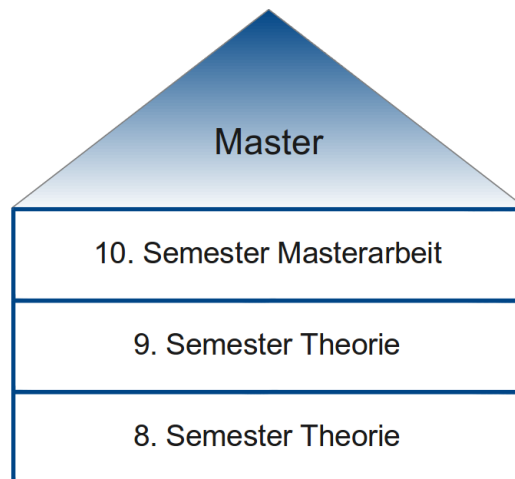
1.2 Studienabschluss

Die Technische Hochschule Ingolstadt verleiht nach erfolgreicher Abschlussprüfung den folgenden akademischen Grad:

Master of Science (M.Sc.)

1.3 Studienaufbau

Das Studium wird als Vollzeitstudium angeboten; die Regelstudienzeit beträgt drei theoretische Studiensemester (90 ECTS-Punkte), wobei das dritte Semester überwiegend der Anfertigung der Masterarbeit dienen soll.



In den beiden ersten Studiensemestern werden Studienschwerpunkte geführt. Zu Beginn des Studiums ist ein Studienschwerpunkt zu wählen.

Ab dem Wintersemester 2014/15 werden ausschließlich die beiden neuen Schwerpunkte *Security and Safety* und *Information Systems Engineering* angeboten.

Weitere Schwerpunkte entstehen u.a. in Kooperation mit anderen bayerischen Hochschulen.

1.4 Qualifikationsvoraussetzungen

Die Qualifikationsvoraussetzungen für die Zulassung zum Masterstudiengang sind wie folgt in der SPO2017 formuliert¹:

1. Qualifikationsvoraussetzung für den Zugang zum Masterstudium ist der Nachweis eines erfolgreichen Abschlusses eines Studiums an einer deutschen Hochschule mit mindestens 210 ECTS-Leistungspunkten oder äquivalentem Studiumumfang im Bereich Informatik oder artverwandten Bereichen oder ein gleichwertiger erfolgreicher in- oder ausländischer Abschluss. Der Nachweis wird durch Vorlage des Abschlusszeugnisses geführt. 3 Über die Gleichwertigkeit

¹rechtlich verbindlich für Vorrückungs- und Zulassungsvoraussetzungen ist nur die SPO Informatik

entscheidet die Prüfungskommission unter Beachtung der Grundsätze des Art. 63 Abs. 1 BayH-SchG.

2. Bei Abschlüssen, die keine Leistungspunkte aufweisen, werden die nachgewiesenen Zeitstunden (Workload) in Leistungspunkte umgerechnet, wobei ein Leistungspunkt einer Stundenbelastung von 30 Zeitstunden entspricht. Falls keine Zeitstunden nachgewiesen werden, werden pro theoretischem Studiensemester 30 ECTS anerkannt. Praxissemester werden mit weiteren 30 ECTS anerkannt soweit diese dem praktischen Studiensemester in Art und Umfang an der technischen Hochschule Ingolstadt entsprechen.
3. Die Nachweise gemäß Abs. 1 sind spätestens am Tage der Immatrikulation zu erbringen. Wird der Nachweis nach Abs. 1 lit. a) Satz 1 nicht bis zum Ende des Bewerbungszeitraums erbracht, ist bis dahin eine entsprechende Bestätigung der Hochschule zu erbringen, die glaubhaft die Erfüllung der Voraussetzung gemäß Absatz 1 lit. a) Satz 1 bis zum Ende des Immatrikulationszeitraums bestätigt; die Pflicht nach Satz 1 bleibt bestehen.
4. Bewerber mit weniger als 210 aber mindestens 180 ECTS-Leistungspunkten werden zugelassen, wenn die übrigen Voraussetzungen nach Abs. 1 erfüllt sind und mit dem Antrag auf Zulassung die fehlenden Kompetenzen nachgewiesen werden. Bei den fehlenden Kompetenzen nach Satz 1 handelt es sich um den Nachweis des Ausgleichs der Kompetenzlücke im Umfang von weiteren bis zu 30 ECTS-Leistungspunkten zu der in Abs. 1 Satz 1 nachzuweisenden Qualifikationsvoraussetzung auf mindestens Bachelorniveau. Die fehlenden Kompetenzen sind spätestens zum Ende des Bewerbungszeitraums nachzuweisen und [...]
5. Bei Nichtzulassung eines Bewerbers wird ihm dies mit einer Begründung schriftlich mitgeteilt.

Näheres regelt die Studien- und Prüfungsordnung des Masterstudiengangs Informatik vom 17.07.2017.

1.5 Fachstudienberatung

Für alle fachlichen Fragen und Probleme im Zusammenhang mit dem Studium steht der Fachstudienberater zur Verfügung:

Prof. Dr. Jörg Hunsinger, Gebäude B, Raum B207, Tel. 0841 / 9348 – 2920

Die während des Semesters geltenden Sprechstunden werden jeweils durch Aushang bekannt gemacht.

1.6 Studiengangleitung

Für Fragen die organisatorische Abwicklung des Studiengangs betreffend, steht der Studiengangleiter zur Verfügung:

Prof. Dr. Franz Regensburger, Gebäude Z, Raum 468, Tel. 0841 / 9348 – 2780

Die während des Semesters geltenden Sprechstunden werden jeweils durch Aushang bekannt gemacht.

2 Curriculare Struktur

Der Masterstudiengang Informatik beginnt jedes Sommer- und jedes Wintersemester. In der Regel werden die einzelnen Module entweder im Sommersemester oder im Wintersemester angeboten.

Die Inhalte der Module des Sommersemesters sind unabhängig von den Inhalten der Module des Wintersemesters und umgekehrt. Dadurch ist gewährleistet, dass der Einstieg in das Masterstudium sowohl im Winter als auch im Sommer möglich ist.

Die Studierenden des ersten und zweiten Semesters nehmen in der Regel gemeinsam an den Veranstaltungen teil. Die folgende Tabelle stellt ein mögliches Curriculum dar, wobei die Module der ersten beiden Semester aus oben genannten Gründen nach Sommer- und Wintersemester gruppiert sind und nicht nach erstem und zweitem Semester.

2.1 Allgemeine Pflichtfächer

Der erste Studienabschnitt umfasst zwei theoretische Semester.

Lfd. Nr.	Fach	Sommersemester		Wintersemester	
		SWS	CP	SWS	CP
	Pflichtfächer				
1	Architektur- und Entwurfsmuster der Softwaretechnik	4	5 (schrP)		
2	Komplexität von Algorithmen und deren Optimierung			4	5 (schrP)
3	Software-Engineering für skalierbare Anwendungen			4	5 (schrP)
4	Angewandte Logik für Modellierung und Verifikation	4	5 (schrP)		
5	Seminar zu Themen der Informatik	2	3 (SV)		
6	Module der Studienschwerpunkte (Details nächste Seite)	12	15	8	10
7	Projekt			4	10 (Proj)
8	Masterarbeit (findet im dritten Semester statt)				
9	Seminar zur Stärkung der Schlüsselqualifikationen	2	2 (SV)		
	Summe	24	30	20	30

Legende:

schrP	schriftliche Prüfung
Proj	Projektarbeit
SV	Seminar mit Vortrag

Der zweite Studienabschnitt umfasst das dritte Semester, in dem in der Regel die Master-Arbeit angefertigt wird.

Lfd. Nr.	Fach	Sommersemester	
		SWS	CP
	Pflichtfächer		
8	Masterarbeit	1	30 (MA)
	Summe	1	30

Legende:

MA Masterarbeit

2.2 Studienschwerpunkte

Ab dem Wintersemester 2014/15 werden ausschließlich die beiden neuen Schwerpunkte *Security and Safety* und *Information Systems Engineering* angeboten.

Beide Schwerpunkte umfassen jeweils fünf Module, die im ersten Studienabschnitt (erstes und zweites Semester) zu belegen sind.

Lfd. Nr.	Fach	Sommersemester		Wintersemester	
		SWS	CP	SWS	CP
Schwerpunkt: Security and Safety					
6.1.1	Sicherheit moderner Netzwerke	4	5 (schrP)		
6.1.2	Security-Engineering in der IT	4	5 (mdIP)		
6.1.3	Computer-Forensik	4	5 (prA)		
6.1.4	Normen und Prozesse zur Entwicklung sicherheitskritischer Systeme			4	5 (schrP)
6.1.5	Software-Technik für sicherheitskritische Systeme			4	5 (mdIP)
	Summe	12	15	8	10

Lfd. Nr.	Fach	Sommersemester		Wintersemester	
		SWS	CP	SWS	CP
Schwerpunkt: Information Systems Engineering					
6.2.1	Daten-Management und -Analyse	4	5 (mdIP)		
6.2.2	Hochleistungs-Datenhaltungs-Systeme	4	5 (schrP)		
6.2.3	Enterprise-Architecture-Management	4	5 (mdIP)		
6.2.4	IT-Integrations- und Migrationstechnologien			4	5 (schrP)
6.2.5	Implementierung von Informationssystemen			4	5 (prA)
	Summe	12	15	8	10

Legende:

schrP schriftliche Prüfung
mdIP mündliche Prüfung
prA praktische Arbeit

3 Modulbeschreibungen

3.1 Allgemeine Pflichtmodule

Architektur- und Entwurfsmuster der Softwaretechnik			
Modulkürzel:	IM_AES	SPO-Nr.:	1
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	1
Modulverantwortliche(r):	Hafenrichter, Bernd		
Sprache:	Deutsch		
Leistungspunkte / SWS:	5 ECTS / 4 SWS		
Arbeitsaufwand:	Kontaktstunden:	47 h	
	Selbststudium:	78 h	
	Gesamtaufwand:	125 h	
Lehrveranstaltungen des Moduls:	1 Architektur- und Entwurfsmuster der Softwaretechnik (IM_AES)		
Lehrformen des Moduls:	IM_AES: SU/Ü - seminaristischer Unterricht/Übung		
Prüfungsleistungen:	1 schrP90 - schriftliche Prüfung, 90 Minuten		
Voraussetzungen gemäß SPO:			
Keine			
Empfohlene Voraussetzungen:			
Grundlegende Erfahrung in der Entwicklung kleinerer Software-Systeme und damit verwandter Technologien (UML, Programmierung, Datenbanken)			
Angestrebte Lernergebnisse:			
Nach dem Besuch des Moduls sind die Studierenden in der Lage, <ul style="list-style-type: none">• die Bedeutung der Software Architektur und deren Einfluss auf den Lebenszyklus einer Software wiederzugeben• Funktionale und Nicht-Funktionale-Anforderung auf verschiedene Ebenen einer Architektur abzubilden• komplexe Software-Architekturen zu entwerfen und umzusetzen• das Prinzip Inversion of Control auf Basis einer objektorientierten Programmiersprache anzuwenden• Anforderung auf ein wohl strukturiertes Software-Design zu übertragen• einen Katalog an Design-Patterns zu beschreiben und können diese auf konkrete Problemstellungen übertragen.• die Vor- und Nachteile der verschiedenen Muster und die Auswirkung auf das Design einzuschätzen• die wichtigsten Bestandteile professioneller Buildumgebungen aufzuzählen			
Inhalt:			
<ul style="list-style-type: none">1. Grundlagen:<ul style="list-style-type: none">○ Definition Software Architektur○ Grundprinzipien2. Struktursicht<ul style="list-style-type: none">○ Komponentenarchitektur			

- Aufbau eines Komponentenframeworks mit Inversion of Control und Dependency Injection
- Quasar
- Diskussion anhand eines Beispiels
- 3. Physische Sicht
 - Verteilungsmuster
 - Verfügbarkeit von verschiedenen Verteilungsmuster
 - Clustering
- 4. Process Sicht
 - Grundlegend Muster
 - Muster für Auftragsverarbeitende Server
 - Diskussion anhand eines Beispiels
- 5. Die logische Sicht
 - Designprinzipien
 - Domain Driven Design
 - Grundlagen
 - Supple Design
 - Maintaining the Modell Integrity
- 6. Design Pattern
 - Erzeugerpattern
 - Strukturpattern
 - Verhaltenspattern
- 7. Enterprise Patterns
 - Logical Unit Of Work
- 8. Entwicklungsumgebung
 - Buildserver
 - Testumgebung
 - Continuos Integration und Delivery

Literatur:

- Johannes Siedersleben: "Moderne Software-Architektur: Umsichtig planen, robust bauen mit Quasar", dpunkt Verlag
- Joachim Goll: Architektur- und Entwurfsmuster der Softwaretechnik: Mit lauffähigen Beispielen in Java, Springer Vieweg
- Robert C. Martin: "Clean Code: A Handbook of Agile Software Craftsmanship", Prentice Hall
- Martin Folwer: "Patterns of Enterprise Application Architecture", Addison Wesley
- Eric J. Evans: "Domain-Driven Design: Tackling Complexity in the Heart of Software", Addison Wesley

Komplexität von Algorithmen und deren Optimierung			
Modulkürzel:	IM_KAO	SPO-Nr.:	2
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	1
Modulverantwortliche(r):	Schmidt, Ulrich		
Sprache:	Deutsch		
Leistungspunkte / SWS:	5 ECTS / 4 SWS		
Arbeitsaufwand:	Kontaktstunden:	47 h	
	Selbststudium:	78 h	
	Gesamtaufwand:	125 h	
Lehrveranstaltungen des Moduls:	2 Komplexität von Algorithmen und deren Optimierung (IM_KAO)		
Lehrformen des Moduls:	IM_KAO: SU/Ü - seminaristischer Unterricht/Übung		
Prüfungsleistungen:	2 schrP90 - schriftliche Prüfung, 90 Minuten		
Voraussetzungen gemäß SPO:			
Keine			
Empfohlene Voraussetzungen:			
Gute Kenntnisse der Programmiersprache Java			
Angestrebte Lernergebnisse:			
Nach erfolgreicher Teilnahme an der Lehrveranstaltung sind die Studierenden in der Lage, Algorithmen hinsichtlich ihrer asymptotischen Komplexität zu klassifizieren und kostenoptimale Algorithmen für parallele Rechnerarchitekturen zu entwickeln und zu implementieren.			
Inhalt:			
<ul style="list-style-type: none">• Asymptotisches Wachstum von Funktionen• Klassifizierung von Funktionen mit Landau- und Hardy-Operatoren• Analyse iterativer Algorithmen• Analyse rekursiver Algorithmen• Klassifikation paralleler Rechnerarchitekturen• Daten- und Task-Parallelismus• Parallele Programmierung mit Java Streams und dem Fork-/Join-API• Parallele Programmierung mit OpenCL und Aparapi			
Literatur:			
<ul style="list-style-type: none">• MILLER, Russ und Laurence BOXER, 2012. <i>Algorithms sequential & parallel</i>. 3. Auflage. ISBN 978-1133366805• RAUBER, Thomas und Gudula RUENGER, 2013. <i>Parallel Programming for Multicore and Cluster Systems</i>. 2. Auflage. ISBN 978-3642378003• PATTERSON, David A. und John L. HENNESSY, 2013. <i>Computer Organization and Design</i>. 5. Auflage. ISBN 978-0124077263• HENNESSY, John L. und David A. PATTERSON, 2011. <i>Computer Architecture - A Quantitative Approach</i>. 5. Auflage. ISBN 978-0123838728• MUNSHI, Aaftab und andere, 2011. <i>OpenCL Programming Guide</i>. 1. Auflage. ISBN 978-0321749642			

- Kaeli, David und andere, 2015. *Heterogeneous Computing with OpenCL 2.0*. 3. Auflage. ISBN 978-0128014141

Software-Engineering für skalierbare Anwendungen			
Modulkürzel:	IM_SESA	SPO-Nr.:	3
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	1
Modulverantwortliche(r):	Regensburger, Franz		
Sprache:	Deutsch		
Leistungspunkte / SWS:	5 ECTS / 4 SWS		
Arbeitsaufwand:	Kontaktstunden:	47 h	
	Selbststudium:	78 h	
	Gesamtaufwand:	125 h	
Lehrveranstaltungen des Moduls:	3 Software-Engineering für skalierbare Anwendungen (IM_SESA)		
Lehrformen des Moduls:	IM_SESA: SU/Ü - seminaristischer Unterricht/Übung		
Prüfungsleistungen:	3 schrP90 - schriftliche Prüfung, 90 Minuten		
	Für die Vorlesung sind folgende Grundlagen wünschenswert <ul style="list-style-type: none">Theoretische Grundlagen und praktische Erfahrung im Umgang mit klassischen relationalen Datenbanksystemen wie etwa PostgreSQL oder MySQLGrundkenntnisse in der Anwendung von Methoden der deskriptiven und induktiven Statistik (Hypothesentests, Regressionsanalyse, Normalverteilung, Modelldiagnose). Bei Bedarf werden oben erwähnte Grundlagen kurz in der Vorlesung wiederholt. Es wird jedoch davon ausgegangen, dass die Teilnehmer eventuell bestehende Kenntnislücken selbständig schließen. Die dazu erforderliche Literatur kann der Modulbeschreibung entnommen werden.		
Voraussetzungen gemäß SPO:			
Keine			
Empfohlene Voraussetzungen:			
Aus dem bereits absolvierten Bachelor-Studiengang sollten folgende Grundlagen vorhanden sein: Relationale Datenbanksysteme, Netzwerktechnik (TCP/IP) Stack, Client/Server-Modell, Programmiersprachen (Java, Python), Revision Control (Mercurial oder Git), Statistik (Lineare Regressionsrechnung, Verteilungen, Hypothesentests), Grundkenntnisse in Linux (Installation und Konfiguration von Programmen, Arbeiten auf der Kommandozeile).			
Angestrebte Lernergebnisse:			
Nach dem Besuch des Moduls <ul style="list-style-type: none">kennen die Studierenden grundlegende Technologien, die die Basis bilden für Datenhaltungs- und Analysysteme, welche Datenmengen jenseits der Terabyte-Grenze speichern und bearbeiten können.verstehen die Studierenden, dass derartig große Datenmengen mit ausschließlich vertikal skalierenden Systemen nicht beherrscht werden können, und lediglich horizontal skalierende Ansätze erfolgversprechend sind.kennen die Studierenden unterschiedliche Ausprägungen des verteilten Rechnens, kennen deren wesentliche Eigenschaften und können die Stärken- und Schwächen der einzelnen Varianten benennen.sind die Studierenden in der Lage, abhängig von der Problemstellung, eine geeignete Ausprägung des verteilten Rechnens auszuwählen.			

- kennen die Studierenden verteilte Algorithmen, deren Struktur auf die verteilte Speichertechnik abgestimmt ist.
- kennen die Studierenden gängige Virtualisierungstechniken, welche die Basis für eine wirtschaftliche Umsetzung skalierbarer System bilden.
- verstehen die Studierenden, dass der Einsatz von noSQL-Systemen zwar eine Steigerung des beherrschbaren Datenvolumens mit sich bringt, dass aber im Gegenzug damit meist auch Abstriche hinsichtlich gewohnter Eigenschaften von RDBMS-Systemen (ACID) hingenommen werden müssen.

Inhalt:

- Eine Taxonomie für den Begriff 'Big Data'
- Grundbegriffe im Kontext der Parallelisierung
- Vertikale und Horizontale Skalierung
- Cluster-Computing mit GEARMAN im PC-Pool mit bis zu 24 Knoten
- Konzepte und Implementierung verteilter Wide Column Stores am Beispiel der Peer-to-Peer Datenbank Cassandra
 - Consistent Hashing
 - Unterschiedliche Konsistenzmodelle
 - Probabilistic Accrual Failure Detection
 - Replikationsstrategien, Anti-Entropie, Selbstheilung
 - praktisches Arbeiten im PC-Pool mit einem Cassandra Cluster mit bis zu 24 Knoten
- Cluster-Computing mit Hadoop 2.x
 - Das Hadoop-Ökosystem (HDFS, YARN, MapReduce, Pig, Hive)
 - Der Wide Columns Store HBase im Vergleich zu Cassandra
 - Arbeiten mit einem Hadoop-Cluster im PC-Pool mit bis zu 24 Knoten mit jeweils 6 Cores

Literatur:

- REDMOND, Eric und Jim R. WILSON, 2013. *Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement*. 2. Auflage. ISBN 978-1-93435-692-0
- SEDLMEIER, Peter und Frank RENKEWITZ, 2013. *Forschungsmethoden und Statistik*. ISBN 978-3-8689-4131-9
- ELMASRI, Ramez A. und Shamkant B. NAVATHE, 2009. *Grundlagen von Datenbanksystemen*. 3. Auflage.
- UNTERSTEIN, Michael und Günter MATTHIESSEN, 2012. *Relationale Datenbanken und SQL in Theorie und Praxis*. 5. Auflage.

Angewandte Logik für Modellierung und Verifikation			
Modulkürzel:	IM_ALMV	SPO-Nr.:	4
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	1
Modulverantwortliche(r):	Regensburger, Franz		
Sprache:	Deutsch		
Leistungspunkte / SWS:	5 ECTS / 4 SWS		
Arbeitsaufwand:	Kontaktstunden:	47 h	
	Selbststudium:	78 h	
	Gesamtaufwand:	125 h	
Lehrveranstaltungen des Moduls:	4 Angewandte Logik für Modellierung und Verifikation (IM_ALMV)		
Lehrformen des Moduls:	IM_ALMV: SU/Ü - seminaristischer Unterricht/Übung		
Prüfungsleistungen:	4 schrP90 - schriftliche Prüfung, 90 Minuten		
	Es werden explizit keine Grundkenntnisse auf dem Gebiet der mathematischen Logik vorausgesetzt. Hilfreich aber nicht notwendig sind Grundkenntnisse in funktionaler Programmierung, vorzugsweise in Haskell.		
Voraussetzungen gemäß SPO:			
Keine			
Empfohlene Voraussetzungen:			
Aus dem bereits absolvierten Bachelor-Studium sollten solide mathematische Grundlagen vorhanden sein, insbesondere auf dem Gebiet der naiven Mengenlehre und der diskreten Mathematik (Kombinatorik, Mengen und Relationen). Desweiteren sind Grundlagen der Verbandstheorie (induktive Mengen, Fixpunktsatz von Knaster-Tarski) hilfreich. Für das Verständnis des Lambda-Kalküls und der Logik höherer Stufe sind Grundkenntnisse in der funktionalen Programmierung (ML, Haskell, Scala) von Vorteil.			
Angestrebte Lernergebnisse:			
Nach erfolgreicher Teilnahme an der Lehrveranstaltung sind die Studierenden in der Lage, <ul style="list-style-type: none">• in Grundzügen wichtige formale Systeme der mathematischen Logik sowie deren wesentliche Begrifflichkeiten wiederzugeben.• zu erkennen, dass diese formalen Systeme eine solide Basis für die Modellierung von Soft- und Hardware-Systemen darstellen.• formale Beweise im Kalkül des natürlichen Schließens sowohl manuell als auch unter Benutzung des taktischen Theorembeweisers Isabelle zu führen.• den Lambda-Kalkül zu beschreiben, welcher die Basis für getypte und ungetypte funktionale Programmiersprachen ist• Algorithmen in Isabelle/HOL zu modellieren, deren Eigenschaften zu verifizieren und in funktionale Programme zu überführen			
Inhalt:			
<ul style="list-style-type: none">• Grundbegriffe der mathematischen Logik am Beispiel der Aussagenlogik<ul style="list-style-type: none">○ Syntax: Formelsprache○ Semantik: Interpretation, Modell, Erfüllbarkeit, Gültigkeit			

- Der Kalkül des Natürlichen Schließen von Gerhard Gentzen PNI/PNK (Fragmente für die Aussagenlogik)
- Korrektheit und Vollständigkeit
- Formalisierung der Kalküle PNI/PNK in Isabelle
- Prädikatenlogik erster Stufe
 - Syntax: Formelsprache, Substitution
 - Semantik: Individuenbereiche, Variablenbelegungen, Quantoren
 - Der Kalkül des Natürlichen Schließen von Gerhard Gentzen NI/NK
 - Formalisierung der Kalküle NI/NK in Isabelle
- Der ungetypte Lambda-Kalkül
- Der getypte Lambda-Kalkül
- Grundbegriffe der Logik höherer Stufe (HOL), Unvollständigkeit
- Isabelle/HOL
- Modellierung und Verifikation von Software-Systemen mit Isabelle/HOL

Literatur:

- NIPKOW, Tobias, KLEIN, Gerwin. *Concrete Semantics* [online]. PDF e-Book. Verfügbar unter: <http://www.concrete-semantics.org/>.
- EBBINGHAUS, FLUM und THOMAS, 1992. *Einführung in die mathematische Logik*. 3. Auflage. ISBN 3-411-15603-1
- HINDLEY, Roger und J.P. SELDIN, 2008. *Lambda-Calculus and Combinators: An Introduction*. 2. Auflage. ISBN 978-0521898850

Seminar zu Themen der Informatik

Modulkürzel:	IM_SEMI	SPO-Nr.:	5
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	1
Modulverantwortliche(r):	Regensburger, Franz		
Sprache:	Deutsch		
Leistungspunkte / SWS:	3 ECTS / 2 SWS		
Arbeitsaufwand:	Kontaktstunden:	24 h	
	Selbststudium:	51 h	
	Gesamtaufwand:	75 h	
Lehrveranstaltungen des Moduls:	5 Seminar zu Themen der Informatik (IM_SEMI)		
Lehrformen des Moduls:	IM_SEMI: S - Seminar		
Prüfungsleistungen:	5 SA/P - Seminararbeit mit Präsentation		
Voraussetzungen gemäß SPO:			
Keine			
Empfohlene Voraussetzungen:			
Im Rahmen der Seminarvorbereitung muss einschlägige Fachliteratur bearbeitet werden, die meistens in englischer Sprache vorliegt. Daher müssen die Teilnehmer in der Lage sein, englische Fachtexte aus dem Bereich der Informatik lesen zu können.			
Angestrebte Lernergebnisse:			
Nach dem Besuch des Moduls <ul style="list-style-type: none">haben die Studierenden ihre Fähigkeit vertieft, sich selbständig spezielle fachliche Kenntnisse zu erarbeiten (Literaturarbeit, Analyse, Schlussfolgerungen) und können diese mithilfe des Einsatzes geeigneter Medien nachvollziehbar im Rahmen eines mündlichen Vortrags präsentierensind die Studierenden in der Lage, einer fachlichen Präsentation kritisch zu folgen und die Inhalte mit dem Vortragenden fachlich zu diskutieren (Stärkung der kommunikativen Kompetenz)haben die Studierenden ihre überfachlichen und kommunikativen Kompetenzen verstärktkönnen die Studierenden den Inhalt ihrer Präsentation in Form einer kurzen schriftlichen Ausarbeitung darstellen			
Inhalt:			
Pro Semester werden im Allgemeinen mehrere Seminare angeboten. Gegenstand ist jeweils ein Themenfeld aus der aktuellen Forschung und Entwicklung im Kontext der angebotenen Studienschwerpunkte. Der jeweilige Dozent stellt eine Sammlung von Papieren oder Bücher aus der Fachliteratur zusammen, die zugleich die Basisliteratur für die Vorträge darstellt. Im Zuge des Seminars muss jeder Teilnehmer <ul style="list-style-type: none">Literaturrecherchen durchführeneine Präsentation über sein Thema ausarbeiten und diese im Rahmen einer Doppelstunde mündlich vortrageneine schriftliche Ausarbeitung über das bearbeitete Thema erstellen Detaillierte Hinweise zu Terminen und seine Erwartungen hinsichtlich Inhalt und Umfang der Präsentationen sowie der schriftlichen Ausarbeitung kommuniziert der jeweilige Dozent zu Beginn des Semesters			

Literatur:

Wird vom jeweiligen Dozenten spätestens zu Beginn der Veranstaltung bekannt gegeben.

Projekt			
Modulkürzel:	IM_PRJ	SPO-Nr.:	7
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	1
Modulverantwortliche(r):	Studiengangleiter/in		
Sprache:	Deutsch		
Leistungspunkte / SWS:	10 ECTS / 4 SWS		
Arbeitsaufwand:	Kontaktstunden:	47 h	
	Selbststudium:	203 h	
	Gesamtaufwand:	250 h	
Lehrveranstaltungen des Moduls:	7 Projekt (IM_PRJ)		
Lehrformen des Moduls:	IM_PRJ: Prj - Projekt		
Prüfungsleistungen:	7 prA - praktische Arbeit/Studienarbeit		
	Aus dem einschlägigen Bachelor-Studium werden Kenntnisse auf dem Gebiet Projektmanagement vorausgesetzt.		
Voraussetzungen gemäß SPO:			
Keine			
Empfohlene Voraussetzungen:			
Im Rahmen des Projekts werden in der Regel Software-Komponenten oder ganze Anwendungen entwickelt. Daher sind solide Grundlagen auf folgenden Gebieten erforderlich: Programmierung (Java, Python, eventuell C/C++), Web-Programmierung (Java, PHP, ECMA), Datenbanksysteme, Netzwerktechnik, Revision Control (Git, Mercurial). Desweiteren werden Grundkenntnisse auf dem Gebiet des agilen Projektmanagements (Scrum, Kanban) vorausgesetzt.			
Angestrebte Lernergebnisse:			
Nach dem Besuch des Moduls			
<ul style="list-style-type: none">haben die Studierenden weitere praktische Erfahrungen hinsichtlich der Anwendung von Projektmanagementmethoden gesammelt.können die Studierenden versiert mit Werkzeugen umgehen, die im Rahmen der Durchführung eines IT-Projekts zur Anwendung kommenhaben die Studierenden ihre Fähigkeit ausgebaut, mit fachlichen und nicht-fachlichen Problemen umzugehen, die während der Durchführung eines mehrwöchigen Projekts auftreten könnenhaben die Studierenden ihre Fähigkeit ausgebaut, eine komplexe fachliche Aufgabenstellung zu analysieren und über ein Semester hinweg in einem Team erfolgreich zu bearbeitenkönnen die Studierenden in unterschiedlicher aber stets angemessener Ausführlichkeit über den Projektfortschritt in mündlicher und/oder schriftlicher Form berichtenhaben die Studierenden gelernt, fachliche und nicht-fachliche (insbesondere auch unternehmerische) Ziele des Projekts kritisch zu hinterfragen und im Sinne eines Gesamterfolges des Projekts abzuwägen			
Inhalt:			
Bearbeitung einer semesterbegleitenden Projektaufgabe aus dem Bereich der Informatik in einem Team.			

Im Allgemeinen werden die Projekte in Kooperation mit externen Firmen oder dem hochschuleigenen Forschungszentrum durchgeführt. Alternativ können auch Dozenten gezielt Projektthemen vorgeben, die im Rahmen ihrer Lehr- oder Forschungstätigkeit bearbeitet werden sollen.

Die Projektleitung und die Organisation werden von Studierenden ausgeführt. Der Dozent/Lehrbeauftragte fungiert lediglich als Coach und/oder Auftraggeber.

Als Projektmanagementmethode können klassische Methoden oder agile Methoden wie Scrum oder Kanban verwendet werden. Die Entscheidung darüber, welche Methode verwendet wird, liegt beim Projektteam.

Zu Beginn des Projekts kommuniziert der Dozent/Lehrbeauftragte klar seine Erwartungen hinsichtlich Terminen, Form und Nachweis der individuellen Leistungen, die von allen Studierenden zu erbringen sind.

Das Projektteam einigt sich mit dem Dozenten/Lehrbeauftragten über die Kommunikations- und Dokumentationsformen, die während der Projektlaufzeit von allen Projektteilnehmern (Studierende, Dozent, Auftraggeber) einzuhalten sind.

Zu klären sind:

- Häufigkeit und Dauer von Planungssitzungen
- Art und Durchführung der Treffen (gemeinsam oder virtuell/elektronisch)
- turnusmäßige Treffen (evtl. täglich in Form von Scrum-Meetings etc.)
- Art und Umfang der Deliverables
- Art und Umfang der individuellen Beiträge durch Studierende
- Kriterien für die Beurteilung/Benotung durch den Dozenten

Literatur:

- SHORE, James und Shane WARDEN, 2007. *The Art of Agile Development*. ISBN 978-0596527679
- SCHWABER, Ken und Mike BEEDLE, 2002. *Agile Software Development with Scrum*. ISBN 0-13-207489-3
- REDMOND, Eric und Jim WILSON, 2012. *Seven Databases in Seven Weeks*. ISBN 978-1-93435-692-0

Masterarbeit			
Modulkürzel:	IM_MA	SPO-Nr.:	8
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	3
Modulverantwortliche(r):	Regensburger, Franz		
Sprache:	Deutsch		
Leistungspunkte / SWS:	30 ECTS / 1 SWS		
Arbeitsaufwand:	Kontaktstunden:	12 h	
	Selbststudium:	738 h	
	Gesamtaufwand:	750 h	
Lehrveranstaltungen des Moduls:	8 Masterarbeit (IM_MA)		
Lehrformen des Moduls:	IM_MA: Prj - Projekt		
Prüfungsleistungen:	8 Master-Abschlussarbeit		
	<p>Wichtige Hinweise: Setzen Sie Ihre Betreuer und Erstprüfer regelmäßig in Kenntnis von Ihren Fortschritten. Klären Sie insbesondere deren Erwartungen an den Inhalt der Arbeit ab.</p> <p>Für die Bearbeitung der Masterarbeit wird ein ganzes Semester veranschlagt (30 Leistungspunkte), wohingegen für die Bearbeitung der Bachelorarbeit nur 12 Leistungspunkte veranschlagt werden. Daraus wird ersichtlich, dass hinsichtlich Umfang und Inhalt an eine Masterarbeit wesentlich höhere Ansprüche gestellt werden als an eine Bachelorarbeit.</p> <p>Insbesondere der wissenschaftliche Charakter sollte bei einer Masterarbeit stärker betont werden. Dies betrifft unter anderem:</p> <ul style="list-style-type: none">• Aussagen sollten, wo immer möglich, in den Kontext mit einschlägiger Fachliteratur gestellt werden• Neben herkömmlicher Fachliteratur sollten wesentlich auch Quellen aus der aktuellen Forschung (z.B. Dissertationen und Konferenzbeiträge) einbezogen werden.• Die Arbeitsweise des Absolventen sollte zielgerichtet, methodisch und systematisch sein und explizit in der Abschlussarbeit dokumentiert werden• Quantitative Aussagen, wie etwa Messungen, sollten mit den Mitteln der mathematischen Statistik untersucht und dokumentiert werden.		
Voraussetzungen gemäß SPO:			
Keine			
Empfohlene Voraussetzungen:			
<p>Im Rahmen einer Master-Arbeit empfiehlt es sich, die Befähigung zum wissenschaftlichen Arbeit unter anderem in Form einer soliden statistischen Auswertung zu demonstrieren. Daher sind Grundkenntnisse der deskriptiven und induktiven Statistik von Vorteil. Diesbezügliche Literaturhinweise zum Selbststudium finden Sie bei den Literaturangaben zu diesem Modul.</p>			
Angestrebte Lernergebnisse:			
Nach der erfolgreichen Erstellung der Masterarbeit			

- können die Studierenden ein Problem selbstständig und unter Einsatz wissenschaftlicher Methoden bearbeiten
- können die Studierenden Anforderungen, alternative Lösungsvorschläge sowie möglicherweise die Ausarbeitung einzelner Lösungsansätze bewerten und schriftlich in einer überzeugenden und nachvollziehbaren Weise darstellen
- haben die Studierenden gelernt, eine umfangreiche Aufgabenstellung durch effektives Zeitmanagement in einem vorgegebenen Zeitrahmen zum Abschluss zu bringen

Inhalt:

Eine Masterarbeit ist der wissenschaftliche Abschluss eines Studiums und Bestandteil der Prüfung. Sie soll zeigen, dass der Absolvent in der Lage ist, ein Problem aus seinem Studiengang selbstständig und unter Einsatz wissenschaftlicher Methoden zu bearbeiten.

Studenten erhalten hier die Gelegenheit, selbstständig eine Aufgabe zu bearbeiten, um damit Kreativität, aber auch den Willen und die Befähigung zur Bearbeitung und zum erfolgreichen Abschluss einer gestellten Aufgabe zu zeigen.

Die Erstellung einer Masterarbeit erfordert Können und Wissen auf vier Gebieten:

- Das jeweilige fachliche Wissen, welches zur Bearbeitung des Themas der Masterarbeit benötigt wird
- Techniken, Methoden und Vorgehensweisen des wissenschaftlichen Arbeitens
- Projektmanagement (insbesondere Zeitplanung und Controlling)
- gegebenenfalls Präsentationstechniken

Im Allgemeinen sucht sich der Studierende selbstständig ein Thema für die Abschlussarbeit. Themen werden entweder hochschulintern von Professoren oder wissenschaftlichen Mitarbeitern der Hochschule in Aushängen (auch online) angeboten, oder ergeben sich aus der Kooperation des Studierenden mit einer externen Firma.

Im Fall einer externen Themenstellung muss der Studierende einen Dozenten der Hochschule von seinem Thema begeistern, damit dieser die Rolle des Erstprüfers übernimmt. Zu diesem Zweck empfiehlt es sich, die Themenstellung und die geplante Herangehensweise in einer kurzen Ausarbeitung zu skizzieren. Dieses Exposé dient dazu, den als Erstprüfer gewünschten Dozenten zu überzeugen.

Literatur:

- WEWEL, Max, 2014. *Statistik im Bachelor-Studium der BWL und VML*. 3. Auflage. ISBN 978-3-86894-220-0
- SEDLMEIER, Peter und Frank RENKEWITZ, 2013. *Forschungsmethoden und Statistik*. 2. Auflage. ISBN 978-3-8689-4131-9
- HATZINGER, Reinhold und andere, 2014. *R Einführung durch angewandte Statistik*. 2. Auflage. ISBN 978-3-86894-250-7

Seminar zur Stärkung der Schlüsselqualifikationen			
Modulkürzel:	IM_SSQ	SPO-Nr.:	9
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	1
Modulverantwortliche(r):	Regensburger, Franz		
Sprache:	Deutsch		
Leistungspunkte / SWS:	2 ECTS / 2 SWS		
Arbeitsaufwand:	Kontaktstunden:	24 h	
	Selbststudium:	26 h	
	Gesamtaufwand:	50 h	
Lehrveranstaltungen des Moduls:	9 Seminar zur Stärkung der Schlüsselqualifikationen (IM_SSQ)		
Lehrformen des Moduls:	IM_SSQ: S - Seminar		
Prüfungsleistungen:	9 SA/P - Seminararbeit mit Präsentation		
Voraussetzungen gemäß SPO:			
Keine			
Empfohlene Voraussetzungen:			
Keine			
Angestrebte Lernergebnisse:			
Nach dem Besuch des Moduls haben die Studierenden <ul style="list-style-type: none">• erkannt, dass für das erfolgreiche Agieren in Teams und Organisationsstrukturen über rein fachliches Können hinaus vor allem Kompetenzen hinsichtlich der sogenannten Soft-Skills eine entscheidende Rolle spielen• ihre persönlichen Handlungskompetenzen im Hinblick auf Soft-Skills in verschiedenen Situation verfeinert			
Inhalt:			
<ul style="list-style-type: none">• Wechselnde Inhalte zu den Soft-Skills<ul style="list-style-type: none">○ Zeitmanagement○ Konfliktmanagement○ Teamworking○ Rhetorik○ Schlagfertigkeit○ Argumentation○ Verhandlung○ Moderation• Vermittlung von Grundwissen ebenso wie von konkreten praktischen Tipps, Tricks und Techniken			
Literatur:			
<ul style="list-style-type: none">• VIGENSHOW, Uwe, Björn SCHNEIDER und Ines MEYROSE, 2014. <i>Soft Skills für Softwareentwickler: Frage-techniken, Konfliktmanagement, Kommunikationstypen und -modelle</i>. 3. Auflage. ISBN 978-3-86490-190-4			

- VIGENSCHOW, Uwe, Björn SCHNEIDER und Ines MEYROSE, 2016. *Soft Skills für IT-Führungskräfte und Projektleiter: Softwareentwickler führen und coachen, Hochleistungsteams aufbauen*. 3. Auflage. ISBN 978-3864903953
- HANISCH, Horst, 2017. *Soft Skills-Knigge 2100: Soziale Kompetenz, Persönlichkeit, Selbstmanagement*. 3. Auflage. ISBN 978-3743197084

3.2 Pflichtmodule Schwerpunkt "Security and Safety"

Sicherheit moderner Netzwerke			
Modulkürzel:	IM_SMN	SPO-Nr.:	6.1.1
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	1
Modulverantwortliche(r):	Göldner, Ernst-Heinrich		
Sprache:	Deutsch		
Leistungspunkte / SWS:	5 ECTS / 4 SWS		
Arbeitsaufwand:	Kontaktstunden:	47 h	
	Selbststudium:	78 h	
	Gesamtaufwand:	125 h	
Lehrveranstaltungen des Moduls:	6.1.1 Sicherheit moderner Netzwerke (IM_SMN)		
Lehrformen des Moduls:	IM_SMN: SU/Ü - seminaristischer Unterricht/Übung		
Prüfungsleistungen:	6.1.1 schrP90 - schriftliche Prüfung, 90 Minuten		
Voraussetzungen gemäß SPO:			
Keine			
Empfohlene Voraussetzungen:			
Besuch der Vorlesung und des Praktikums Rechnernetze (Voraussetzung), empfehlenswert ist der Besuch des FW-Fachs "Internet - Netz der Netze"			
Angestrebte Lernergebnisse:			
<p>Nach erfolgreicher Teilnahme an der Lehrveranstaltung sind die Studierenden in der Lage,</p> <ul style="list-style-type: none">• die prinzipielle Struktur des Internets zu beschreiben und beherrschen das Architekturmodell der Kommunikation (TCP/IP-Schichtenmodell) im Detail, die Grundlagen der Vermittlungstechniken und sie können Performance und Wartezeiten der Paketvermittlung abschätzen.• die Eigenschaften und das Verhalten des TCP-Protokolls im prinzipiellen Detail zu beschreiben.• die Architektur eines Routers darzustellen und kennen typische Maßnahmen zur Erhöhung der Ausfallsicherheit in IP-Netze.• die generelle Struktur der heutigen öffentlichen IP-Netze mit allen Technologie-Schichten und deren prinzipiellen Funktionen, Netzelementen und Sicherheitsaspekten zu benennen.• typische Angriffsszenarien auf Rechnernetze und Schutzmaßnahmen dagegen aufzuzählen.• die prinzipiellen Funktionen der Protokolle zum Schutz einer Kommunikation im Internet zu beschreiben.• die Entwicklung, Architektur und Funktion der aktuellen Mobilfunk-Generationen für Sprache und Daten mit speziellem Blick auf die relevanten Sicherheitsfunktionen wiederzugeben.			
Inhalt:			
<p>9. Grundlagen des Internets und der Paketvermittlung:</p> <ul style="list-style-type: none">○ Architekturmodell, Dienste- und Schichtenkonzept○ Grundlagen der Paketvermittlung und der Performance○ Vertiefung des Transportprotokolls TCP: Flusskontrolle und Überlastkontrolle○ Typische Angriffe im Internet <p>10. Struktur des Internets (Festnetz):</p>			

- Anforderungen an den sicheren Betrieb eines Kommunikationsnetzes
 - Netzzugang (Access): Technologien (DSL, Fttx, HFC, typische Übertragungsmedien); Entwicklung des Internet-Access für Privatkunden
 - Struktur aktueller Router und Maßnahmen für den sicheren Betrieb
 - Elemente und Funktionen des optischen Transportnetzes (WDM-Netz)
 - Elemente und Funktionen des synchronen TDM-Transportnetzes (SDH) inkl. Funktionen zur Ersatzschaltung
 - Struktur des IP-Backbones (nationales Netz) und Beispiele für die aktuellen Entwicklungen der Netze (z.B. Virtualisierung, SDN, neue Netzkonzepte)
11. Grundlagen der Sicherheit im Internet:
- Typische Attacken und Schutzmaßnahmen vor Angreifern
 - Operative Sicherheit / Netz Sicherheits-Architektur (Firewalls, Gateways, IDS,...)
 - Protokolle zur sichern Kommunikation (SSL / TLS, IPSec)
12. Mobilfunknetze
- Haupteigenschaften der Mobilfunknetze
 - Historische Entwicklung, genutzte Frequenzen und Spektrum
 - GSM / 2. Generation: Luftschnittstelle, Sprachkodierung, Netzarchitektur
 - Identitäten (SIM-Karte) und Verschlüsselung, Authentisierung, Angriffsvektoren
 - Datendienste im GSM: GPRS/Edge
 - UMTS / 3.Generation: Luftschnittstelle mit WCDMA, Sicherheitsarchitektur im 3G-Netz, schnelle Daten im 3G-Netz (HSPA)
 - LTE / 4. Generation: Luftschnittstelle / LTE-Übertragungstechnik, Netzstruktur (evolved Packet Core)
 - Dienste im all-IP-Netz: IMS

Literatur:

- ECKERT, Claudia, 2014. *IT-Sicherheit: Konzepte - Verfahren - Protokolle*. 9. Auflage. München: De Gruyter Oldenbourg. ISBN 978-3-486-77848-9; 978-3-486-85916-4; 978-3-11-039910-3
- SCHAEFER, Günter und Michael ROßBERG, 2014. *Netzicherheit: Grundlagen & Protokolle; mobile & drahtlose Kommunikation; Schutz von Kommunikationsinfrastrukturen*. 2. Auflage. Heidelberg: dpunkt-Verl. ISBN 978-3-86490-115-7; 3-86490-115-4
- SORGE, Christoph, Nils GRUSCHKA und Luigi LO IACONO, 2013. *Sicherheit in Kommunikationsnetzen*. 1. Auflage. München: Oldenbourg. ISBN 978-3-486-72016-7; 3-486-72016-3; 978-3-486-72017-4
- SAUTER, Martin, 2016. *Grundkurs Mobile Kommunikationssysteme: LTE-Advanced, UMTS, HSPA, GSM, GPRS, Wireless LAN und Bluetooth*. 6. Auflage. Wiesbaden: Springer Fachmedien. ISBN 978-3-658-08342-7; 978-3-658-08341-
- SPITZ, Stephan, Michael PRAMATEFTAKIS und Joachim SWOBODA, 2011. *Kryptographie und IT-Sicherheit: Grundlagen und Anwendungen*. 2. Auflage. Wiesbaden: Vieweg + Teubner. ISBN 978-3-8348-1487-6 ; 978-3-8348-8120-5

Security-Engineering in der IT			
Modulkürzel:	IM_SEIT	SPO-Nr.:	6.1.2
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	1
Modulverantwortliche(r):	Hof, Hans-Joachim		
Sprache:	Deutsch		
Leistungspunkte / SWS:	5 ECTS / 4 SWS		
Arbeitsaufwand:	Kontaktstunden:	47 h	
	Selbststudium:	78 h	
	Gesamtaufwand:	125 h	
Lehrveranstaltungen des Moduls:	6.1.2 Security-Engineering in der IT (IM_SEIT)		
Lehrformen des Moduls:	IM_SEIT: SU/Ü - seminaristischer Unterricht/Übung		
Prüfungsleistungen:	6.1.2 mdP - mündliche Prüfung 30 Minuten		
Voraussetzungen gemäß SPO:			
Keine			
Empfohlene Voraussetzungen:			
Vertiefte Kenntnisse in den Bereichen Betriebssysteme, Netzwerke, Programmierung, Software Engineering sowie Grundkenntnisse IT-Sicherheit			
Angestrebte Lernergebnisse:			
<p>Nach dem Besuch des Moduls</p> <ul style="list-style-type: none">• verfügen die Studierenden über grundlegende Kenntnisse zu Designprinzipien sicherer IT-Systeme, insbesondere unter Berücksichtigung moderner verteilter Systeme.• entwickeln Studierende einerseits ein Verständnis dafür, wie sich Systeme unter Einsatz moderner Virtualisierungstechniken, spezieller Hardware sowie geeigneten Maßnahmen bei Einsatz moderner Betriebssysteme härten lassen.• erlangen die Studierenden durch die Veranstaltung andererseits vertiefte Kenntnisse darüber, welche Techniken des Softwareengineering im besonderen Maße auf die Sicherheit aktueller Software abzielen und wie sicherheitsrelevante Schnittstellenrisiken vermieden werden.			
Inhalt:			
<ul style="list-style-type: none">• Sichere Software Entwicklung (Benutzereingaben, Privilegien, Protokolle)• Absicherung von Kommunikationswegen und Schnittstellen• Verschlüsselung, Algorithmen zum Schlüsselaustausch, Einsatz von Zertifikaten• Systemhärtung auf Betriebesebene• Datensicherheit (Privilegien Filesystem, ACLs)• Sicherheit bei Multi-Thier-Systemen• Absicherung von Datenbanken und Webfrontends• Virtualisierungstechniken, Sandbox• Updatestrategien• interne Sicherheits-Audits, Pentests und Security Patching• Sicherheitsfunktionen moderner Software• Code Obfuscation			

- Techniken zur Authentifizierung und Identifizierung
- Backupstrategien unter Sicherheitsgesichtspunkten, sichere Datenaufbewahrung

Literatur:

- Security Engineering: A Guide to Building Dependable Distributed Systems, Ross J. Anderson, John Wiley & Sons; Auflage: 2, 2008.
- Cyber Security Engineering: A Practical Approach for Systems and Software Assurance von Nancy R. Mead (Autor), Carol Woody (Autor), Pearson Education, 2016.
- Basiswissen Sichere Software: Aus- und Weiterbildung zum ISSECO Certified Professional for Secure Software Engineering, Sachar Paulus, Dpunkt Verlag, 2011.

Computer-Forensik			
Modulkürzel:	IM_CF	SPO-Nr.:	6.1.3
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	1
Modulverantwortliche(r):	Hahndel, Stefan		
Sprache:	Deutsch		
Leistungspunkte / SWS:	5 ECTS / 4 SWS		
Arbeitsaufwand:	Kontaktstunden:	47 h	
	Selbststudium:	78 h	
	Gesamtaufwand:	125 h	
Lehrveranstaltungen des Moduls:	6.1.3 Computer-Forensik (IM_CF)		
Lehrformen des Moduls:	IM_CF: SU/Ü - seminaristischer Unterricht/Übung		
Prüfungsleistungen:	6.1.3 prA - praktische Arbeit/Studienarbeit		
Voraussetzungen gemäß SPO:			
Keine			
Empfohlene Voraussetzungen:			
Grundlagen der Programmierung/Programmierkenntnisse; Grundkenntnisse Betriebssysteme; Grundkenntnisse IT-Sicherheit			
Angestrebte Lernergebnisse:			
<p>Nach erfolgreicher Absolvierung des Moduls sind die Studierenden in der Lage,</p> <ul style="list-style-type: none">theoretische Grundlagen der Computer-Forensik und entsprechende Prozessmodelle basierend auf Carrier's Hypothesen-basierten Ansatz und seinem Referenzmodell für Daten auf Dateisystemen wiederzugeben.wichtige Angriffsmuster auf Computersysteme zu beschreiben und wissen, welche Spuren diese hinterlassen.Dateisysteme einer forensischen Analyse zu unterziehen.die wichtigsten Methoden zu Netzwerk-/Internet-Forensik und Malware Analyse zu beschreiben.			
Inhalt:			
<ul style="list-style-type: none">Methoden von Angreifern und typische AngriffsmusterProzessmodelle für Forensic ComputingTechnologie moderner Speichersysteme: Harddisk, SSDs, DRam, Flash, MRams etc.Disk Volumes und Partitionen im DetailDiverse Dateisysteme, Verfahren zur Wiederherstellung von Daten (FAT, NTFS und Unix/Linux-Dateisysteme)Netzwerk und Internet-Forensik: z.B. Aufspüren von HTTP-Requests und EmailsFortgeschrittene Werkzeuge zur Computer-ForensikUmgang mit verschlüsselten Daten, Aufspüren von VerschlüsselungGrundlagen der Multimedia-Forensik (Analyse von Bild- und Audiodaten)Fortgeschrittene Carvingtechniken			

Literatur:

- GESCHONNECK, Alexander, 2014. *Computer-Forensik: Computerstraftaten erkennen, ermitteln, aufklären*. 6. Auflage. ISBN ISBN-13: 978-3864901331
- KUHLEE, Lorenz und Victor VOELZOW, 2012. *Computer-Forensik Hacks*. 1. Auflage.
- WILLER, Christoph, 2012. *PC-Forensik*. ISBN ISBN-13: 978-3936546606
- DEWALD, Andreas und Felix FREILING, 2015. *Forensische Informatik*. 3. Auflage. ISBN ISBN-13: 978-3842379473
- CARRIER, Brian, 2005. *File System Forensic Analysis*. ISBN 978-0321268174
- SIKORSKI, Michael und Andrew HONIG . *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*. 1. Auflage. ISBN ISBN-13: 978-1593272906

Normen und Prozesse zur Entwicklung sicherheitskritischer Systeme			
Modulkürzel:	IM_NPES	SPO-Nr.:	6.1.4
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	1
Modulverantwortliche(r):	Gold, Robert		
Sprache:	Deutsch		
Leistungspunkte / SWS:	5 ECTS / 4 SWS		
Arbeitsaufwand:	Kontaktstunden:	47 h	
	Selbststudium:	78 h	
	Gesamtaufwand:	125 h	
Lehrveranstaltungen des Moduls:	6.1.4 Normen und Prozesse zur Entwicklung sicherheitskritischer Systeme (IM_NPES)		
Lehrformen des Moduls:	IM_NPES: SU/Ü - seminaristischer Unterricht/Übung		
Prüfungsleistungen:	6.1.4 schrP90 - schriftliche Prüfung, 90 Minuten		
Voraussetzungen gemäß SPO:			
keine			
Empfohlene Voraussetzungen:			
Software Engineering			
Angestrebte Lernergebnisse:			
<p>Nach erfolgreicher Teilnahme an der Lehrveranstaltung sind die Studierenden in der Lage,</p> <ul style="list-style-type: none">• Normen und Prozesse zur Entwicklung sicherheitskritischer Systeme zu beschreiben, im Detail zu erklären und auf bisher unbekannte Anwendungsfälle anzuwenden<ul style="list-style-type: none">◦ Zuverlässigkeitsanalysen durchführen◦ Die Normen IEC 61508, ISO 26262, ED-12C/DO-178C kennen und anwenden◦ Funktionale Sicherheitskonzepte und Sicherheitsnachweise erstellen◦ Verfahren zur Überwachung und Diagnose von Systemen kennen und anwenden◦ Spezielle Software-Prüfmethoden in der Entwicklung sicherheitskritischer Software kennen und anwenden• Normen und Prozesse zur Entwicklung sicherheitskritischer Systeme in der Software-Architektur, im Software-Design einzusetzen• zu verstehen, dass unsere Gesellschaft zu Recht hohe Ansprüche an das Verhalten von Informatikern stellt, welches stets im Einklang mit den sozialen und ethischen Werten unserer Gesellschaft zu sein hat• aufgrund von Erfahrungen in praktischen Anwendungen zu wissen, wo und wie kommerzielle, soziale und ethische Erwägungen, zum Teil widerstreitend, den Alltag des professionellen Informatikers bestimmen.• nachzuvollziehen, dass die Umsetzung von Sicherheitsrichtlinien nur gewährleistet werden kann, wenn deren Einhaltung in allen Phasen des Lebenszyklus von IT-Systemen explizit beachtet wird - sie können einschlägige Methoden und Engineering-Werkzeuge anwenden, welche dies sicherstellen.• wesentliche Normen und Gesetze, die die Produktsicherheit regeln, zu beschreiben und können deren Einhaltung durch den Einsatz entsprechender Methoden sicherstellen.• Safety-Risikoanalysen von IT-Systemen durchzuführen• mittels der nötigen Grundlagen und einem ausgebildeten Abstraktionsvermögen, sich selbstständig in neue unbekannte Fachgebiete und komplexe Problemstellungen einzuarbeiten			

Inhalt:

- Einführung und Grundbegriffe
 - Sicherheit
 - Zuverlässigkeit und Ausfallrate
 - Zuverlässigkeitswachstumsmodelle
- Gefährdungs- und Risikoanalyse
 - IEC 61508
 - ISO 26262 (Risikoparameter und ASIL, Dekomposition)
- Die automotive Sicherheitsnorm ISO 26262
 - Teile der ISO 26262
 - Sicherheitslebenszyklus
 - Funktionale Sicherheitsanforderungen
 - Sicherheitsnachweis
 - Anforderungen an die Software-Entwicklung
- Die Luftfahrtnorm ED-12C/DO-178C
 - Einführung
 - Software Considerations in Airborne Systems
- Überwachung und Diagnose von Systemen
 - Verfahren zur Fehlererkennung
 - Fehlerbehandlung
 - Ablaufüberwachung (Kontrollflussgraphen, CFCSS, CFEDC)
 - Diversität (Arten von Diversität, Versagenswahrscheinlichkeit)
- Prüfung von Software
 - Überblick
 - Statische Analyse (MISRA-C, abstrakte Interpretation / PolySpace)

Literatur:

- BÖRCSÖK, Josef, 2015. *Funktionale Sicherheit: Grundzüge sicherheitstechnischer Systeme*. 4. Auflage. Berlin; Offenbach: VDE Verlag GmbH. ISBN 978-3-8007-3819-9, 978-3-8007-3590-7
- EHRENBERGER, Wolfgang, 2002. *Software-Verifikation: Verfahren für den Zuverlässigkeitsnachweis von Software*. München [u.a]: Hanser. ISBN 3-446-21624-3
- GEBHARDT, Vera, Gerhard M. RIEGER und Jürgen MOTTOK, 2013. *Funktionale Sicherheit nach ISO 26262: ein Praxisleitfaden zur Umsetzung*. 1. Auflage. Heidelberg: dpunkt-Verl. ISBN 978-3-89864-788-5, 3-89864-788-9
- ROSS, Hans-Leo, 2014. *Funktionale Sicherheit im Automobil: ISO 26262, Systemengineering auf Basis eines Sicherheitslebenszyklus und bewährter Managementsysteme*. München [u.a.]: Hanser. ISBN 978-3-446-43632-9, 3-446-43632-4
- SCHÄUFFELE, Jörg und Thomas ZURAWKA, 2013. *Automotive Software Engineering: Grundlagen, Prozesse, Methoden und Werkzeuge effizient einsetzen*. 5. Auflage. Wiesbaden: Springer Vieweg. ISBN 978-3-8348-2469-1, 978-3-8348-2470-7

Software-Technik für sicherheitskritische Systeme			
Modulkürzel:	IM_STSS	SPO-Nr.:	6.1.5
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	1
Modulverantwortliche(r):	Facchi, Christian		
Sprache:	Deutsch		
Leistungspunkte / SWS:	5 ECTS / 4 SWS		
Arbeitsaufwand:	Kontaktstunden:	47 h	
	Selbststudium:	78 h	
	Gesamtaufwand:	125 h	
Lehrveranstaltungen des Moduls:	6.1.5 Software-Technik für sicherheitskritische Systeme (IM_STSS)		
Lehrformen des Moduls:	IM_STSS: SU/Ü - seminaristischer Unterricht/Übung		
Prüfungsleistungen:	6.1.5 mdIP - mündliche Prüfung 30 Minuten		
Voraussetzungen gemäß SPO:			
Keine			
Empfohlene Voraussetzungen:			
fundierte Programmierkenntnisse, Kenntnisse bei der Modellierung von SW mit UML, Kenntnisse bei der Programmierung von verteilten und auch reaktiven Systemen			
Angestrebte Lernergebnisse:			
Nach erfolgreicher Teilnahme an der Lehrveranstaltung sind die Studierenden in der Lage, <ul style="list-style-type: none">• ihre erworbenen vertieften Kenntnisse im Bereich der Qualitätssicherung (insbesondere bezüglich der Qualitätsaspekte Safety und Security) auf komplexe, sicherheitskritische Software anzuwenden (z.B. eingebettete Systeme).• die Zuordnung von Qualitätssicherungsmaßnahmen zu verschiedenen Phasen des Softwareentwicklungsprozesses wiederzugeben und diese Maßnahmen in konkreten Problemstellungen anzuwenden			
Inhalt:			
<ul style="list-style-type: none">• Grundlagen<ul style="list-style-type: none">○ Grundlagen zur Programmierung sicherheitskritischer Systeme (z.B. Parallele Programmierung, hardwarenahe Programmierung, Echtzeitsysteme)○ Grundlagen zu Modellierungstechniken○ Grundlagen zu IT-Sicherheit• Qualitätssicherung in den Entwicklungsphasen<ul style="list-style-type: none">○ Define○ Design○ Develop○ Deploy○ Maintain• Software Management / Querschnittsprozesse / Security Development Lifecycle• Anwendungsbeispiele			

Literatur:

- LIGGESMEYER, Peter, 2002. *Software-Qualität*.
- PEZZE, Mauro und Michael YOUNG, 2009. *Software testen und analysieren*.
- SCHAEUFFELE, Jörg und Thomas ZURAWKA, 2006. *Automotive Software Engineering*. 3. Auflage.

3.3 Pflichtmodule Schwerpunkt "Information Systems Engineering"

Daten-Management und -Analyse			
Modulkürzel:	IM_DMA	SPO-Nr.:	6.2.1
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	1
Modulverantwortliche(r):	Kaiser, Melanie		
Sprache:	Deutsch		
Leistungspunkte / SWS:	5 ECTS / 4 SWS		
Arbeitsaufwand:	Kontaktstunden:	47 h	
	Selbststudium:	78 h	
	Gesamtaufwand:	125 h	
Lehrveranstaltungen des Moduls:	6.2.1 Daten-Management und -Analyse (IM_DMA)		
Lehrformen des Moduls:	IM_DMA: SU/Ü - seminaristischer Unterricht/Übung		
Prüfungsleistungen:	6.2.1 mdIP - mündliche Prüfung 30 Minuten		
Voraussetzungen gemäß SPO:			
Keine			
Empfohlene Voraussetzungen:			
Relationale Datenbanksysteme, Statistik, Programmierkenntnisse			
Angestrebte Lernergebnisse:			
<p>Nach dem Besuch des Moduls sind die Studierenden in der Lage,</p> <ul style="list-style-type: none">• Technologien und Methoden für die Verwaltung und Auswertung großer Informationsmengen zu beschreiben.• die unterschiedlichen Datenmodelle sowie die Vor- und Nachteile der wesentlichen NoSQL-Datenbanksysteme zu erläutern.• die Eignung der einzelnen Systeme für konkrete Einsatzszenarien zu bewerten.• grundlegende Verfahren der Datenanalyse wiederzugeben und diese zur Beschreibung und Exploration von Datenquellen einzusetzen.• grundlegende statistische Kennwerte zur uni- und bivariaten Beschreibung von Datenstrukturen zu berechnen, erklären und zu interpretieren.• Verteilungen und Zusammenhänge in Daten anhand geeigneter graphischer Darstellungsformen zu visualisieren.• Einsatzgebiete und Funktionsweise grundlegender Verfahren des maschinellen Lernens (z.B. Regression, Entscheidungsbäume, Clusterverfahren) zu erklären.• grundlegende Verfahren der Datenanalyse und des maschinellen Lernens mittels der Programmiersprache Python auf neue Aufgabenstellungen anzuwenden und die Ergebnisse anhand geeigneter Kennzahlen zu bewerten und zu interpretieren.			
Inhalt:			
<p>NoSQL-Datenbanken</p> <ul style="list-style-type: none">• Grundlagen• Datenmodelle			

- Kategorien: Key-Value DBs, Document DBs, Column-Family Stores, Graph DBs
- Einsatzgebiete

Datenanalyse mit Python

- Grundlagen der Datenanalyse
- Grundlagen Python
- Deskriptive Analysen
- Datenvisualisierung
- Modellierung und Machine Learning Algorithmen

Literatur:

- GRUS, Joel, 2015. *Data Science from Scratch: first principles with Python*. 1. Auflage. Beijing [u.a.]: O'Reilly. ISBN 978-1-4919-0142-7
- SADALAGE, Pramod J. und Martin FOWLER, 2013. *NoSQL distilled: a brief guide to the emerging world of polyglot persistence* 1. Auflage. Upper Saddle River, NJ; Munich [u.a.]: Addison-Wesley. ISBN 978-0-321-82662-6
- NELLI, Fabio, 2015. *Python data analytics: data analysis and science using pandas, matplotlib, and the Python programming language*. New York, NY: Apress. ISBN 978-1-4842-0959-2, 978-1-4842-0958-5
- EDLICH, Stefan, 2011. *NoSQL: Einstieg in die Welt nichtrelationaler Web-2.0-Datenbanken*. 2. Auflage. München: Hanser. ISBN 978-3-446-42753-2

Hochleistungs-Datenhaltungs-Systeme			
Modulkürzel:	IM_HDS	SPO-Nr.:	6.2.2
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	1
Modulverantwortliche(r):	Rasch, Jochen		
Sprache:	Deutsch		
Leistungspunkte / SWS:	5 ECTS / 4 SWS		
Arbeitsaufwand:	Kontaktstunden:	47 h	
	Selbststudium:	78 h	
	Gesamtaufwand:	125 h	
Lehrveranstaltungen des Moduls:	6.2.2 Hochleistungs-Datenhaltungs-Systeme (IM_HDS)		
Lehrformen des Moduls:	IM_HDS: SU/Ü - seminaristischer Unterricht/Übung		
Prüfungsleistungen:	6.2.2 schrP90 - schriftliche Prüfung, 90 Minuten		
Voraussetzungen gemäß SPO:			
Keine			
Empfohlene Voraussetzungen:			
Kenntnisse in relationalen Datenbanksystemen und SQL			
Angestrebte Lernergebnisse:			
<p>Nach der erfolgreichen Teilnahme am Modul sind die Studierenden in der Lage, die zentralen Anforderungen an die Leistungsfähigkeit von Datenhaltungs- bzw. Datenbanksystemen als essentieller Basis für moderne Unternehmensanwendungen (wie z.B. ERP-Systeme) zu beschreiben und zu erläutern, ebenso wie die grundlegenden Konzepte zur Sicherstellung der Leistungsfähigkeit dieser Datenbanksysteme.</p> <p>Sie kennen die technischen Grundlagen und Prinzipien spalten-/zeilenbasierter Datenhaltungs-systeme und reiner oder hybrider In-Memory-Datenhaltungssysteme und sind in der Lage, diese hinsichtlich ihrer Wirkungen - sowohl einzeln, als auch im Zusammenspiel - einzuschätzen und einzuordnen. Sie sind mit ausgewählten In-Memory-Datenhaltungssystemen und zentralen Werkzeugen solcher Systeme vertraut.</p>			
Inhalt:			
<ul style="list-style-type: none">• Anforderungen an Hochleistungs-Datenbanksysteme als Basis für Unternehmensanwendungen• Architektur und Arbeitsweise moderner Datenbanksysteme• Konzepte und Prinzipien der Skalierbarkeit und Performance-Optimierung: Indizes, DB-Statistiken, Puffer, Parallelisierung, Kompression• Absicherung von Datenbanksystemen und Datenbanken: Konzepte und Vorgehensweisen für Backup, Restore und Recovery; Gewährleistung von Hochverfügbarkeit und Disaster Recovery• Konzepte, Prinzipien und technische Grundlagen In-Memory-basierter Datenbanksysteme und Verdeutlichung an aktuellen kommerziellen Systemen			
Literatur:			
<ul style="list-style-type: none">• BERG, Bjarne und Penny SILVIA, 2015. <i>Einführung in SAP HANA</i>. 2. Auflage. Bonn: Rheinwerk Publ. ISBN 978-3-8362-3459-7• PLATTNER, Hasso, 2013. <i>Lehrbuch In-Memory Data Management: Grundlagen der In-Memory-Technologie</i>. Wiesbaden: Springer Gabler. ISBN 978-3-658-03212-8			

- PLATTNER, Hasso und Alexander ZEIER, 2012. *In-memory data management: technology and applications* 2. Auflage. Berlin [u.a.]: Springer. ISBN 978-3-642-29574-4

Enterprise-Architecture-Management			
Modulkürzel:	IM_EAM	SPO-Nr.:	6.2.3
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	1
Modulverantwortliche(r):	Rasch, Jochen		
Sprache:	Deutsch		
Leistungspunkte / SWS:	5 ECTS / 4 SWS		
Arbeitsaufwand:	Kontaktstunden:	47 h	
	Selbststudium:	78 h	
	Gesamtaufwand:	125 h	
Lehrveranstaltungen des Moduls:	6.2.3 Enterprise-Architecture-Management (IM_EAM)		
Lehrformen des Moduls:	IM_EAM: SU/Ü - seminaristischer Unterricht/Übung		
Prüfungsleistungen:	6.2.3 mdIP - mündliche Prüfung 30 Minuten		
Voraussetzungen gemäß SPO:			
Keine			
Empfohlene Voraussetzungen:			
Grundkenntnisse über die IT-Organisation in Unternehmen und die beteiligten Parteien, Verständnis für die Herausforderungen des Managements von Informationstechnologie in Unternehmen.			
Angestrebte Lernergebnisse:			
<p>Nach erfolgreicher Teilnahme an den Modulveranstaltungen sind die Studierenden in der Lage,</p> <ul style="list-style-type: none">• die weitschichtigen Problemstellungen einer IT-Organisation bei der Gestaltung und dem Management komplexer IT-Systemlandschaften zu reflektieren.• den Beitrag, den das Enterprise Architecture Management (EAM) hierzu liefert, zu erklären und die Grundprinzipien des EAM anzuwenden. <p>Sie sind mit ausgewählten EAM-Methoden und -Werkzeugen vertraut und geübt, können diese hinsichtlich ihrer Einsetzbarkeit im Unternehmen einschätzen und auf kleinere Problemstellungen der Praxis anwenden. Der Zusammenhang mit anderen Disziplinen ist den Studierenden klar und kann erklärt und an konkreten Handlungssituationen hergestellt werden.</p>			
Inhalt:			
<ul style="list-style-type: none">• IT-Organisation im Unternehmen: Aufgaben, Rollen, Ziele, Zusammenhänge• Metamodelle, Architekturschichten und Architekturprinzipien des EAM• IT-Repository und EAM-Daten• EAM-Visualisierungen• Frameworks (z. B. TOGAF)• IT-Governance, Reifegrade und IT-Prozesse (u. a. Zusammenhänge mit Strategie-/Prozessmanagement sowie mit Software-Engineering und IT-Integration)• Planung der Einführung von EAM, Szenarien• Modellgetriebene Ansätze im Zusammenhang mit EAM (optional)			

Literatur:

- HANSCHKE, Inge, 2016. *Enterprise Architecture Management - einfach und effektiv: ein praktischer Leitfaden für die Einführung von EAM*. 2. Auflage. München: Hanser. ISBN 978-3-446-44724-0, 3-446-44724-5
- KELLER, Wolfgang, 2017. *IT-Unternehmensarchitektur: von der Geschäftsstrategie zur optimalen IT-Unterstützung*. 3. Auflage. Heidelberg: dpunkt.verlag. ISBN 978-3-86490-406-6, 3-86490-406-4
- DESFRAY, Philippe und Gilbert RAYMOND, 2014. *Modeling Enterprise Architecture with TOGAF: A Practical Guide Using UML and BPMN*. Saint Louis: Elsevier Science. ISBN 978-0-12-419995-8, 978-0-12-419984-2

IT-Integrations- und Migrationstechnologien			
Modulkürzel:	IM_ITIM	SPO-Nr.:	6.2.4
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	1
Modulverantwortliche(r):	Hafenrichter, Bernd		
Sprache:	Deutsch		
Leistungspunkte / SWS:	5 ECTS / 4 SWS		
Arbeitsaufwand:	Kontaktstunden:	47 h	
	Selbststudium:	78 h	
	Gesamtaufwand:	125 h	
Lehrveranstaltungen des Moduls:	6.2.4 IT-Integrations- und Migrationstechnologien (IM_ITIM)		
Lehrformen des Moduls:	IM_ITIM: SU/Ü - seminaristischer Unterricht/Übung		
Prüfungsleistungen:	6.2.4 schrP90 - schriftliche Prüfung, 90 Minuten		
Voraussetzungen gemäß SPO:			
Keine			
Empfohlene Voraussetzungen:			
Erfahrung in grundlegenden Technologien (XML, Datenbanken, Rest und Java)			
Angestrebte Lernergebnisse:			
<p>Nach erfolgreicher Teilnahme an der Lehrveranstaltung sind die Studierenden in der Lage,</p> <ul style="list-style-type: none">• die Notwendigkeit von Integration und Migration zu beurteilen• grundlegende Integrationsarten und Integrationsmuster einzuschätzen• die Architektur einer Integrationssoftware und die wesentlichen Bestandteile zu erläutern• Muster für die Integration von Systemen einzuschätzen und zu implementieren• typische Protokolle und Standards aus verschiedenen Industriebereichen einzuordnen und zu erklären• die Integration relevante Standards zu beschreiben• grundlegende Begriffe und die Notwendigkeit der Migration wiederzugeben• Methoden und Techniken der Softwaremigration anzuwenden			
Inhalt:			
<p>13. Einleitung</p> <ul style="list-style-type: none">○ Ausgangssituation○ Heterogene Systemlandschaften <p>14. Grundlagen</p> <ul style="list-style-type: none">○ Integrationsarten○ Integrationsarchitektur <p>15. Enterprise Service Bus</p> <ul style="list-style-type: none">○ Integration Architecture Blueprint○ Enterprise Integration Patterns <p>16. Architektur</p> <ul style="list-style-type: none">○ Kopplungsarchitektur			

- Lose Kopplung in verteilten Systemen
- Transaktionen
- 17. Technische Standards
 - WS-Transaction
 - WS-Security
 - WS-Reliable Messaging
- 18. Industriestandards
 - Gesundheitswesen
 - eCommerce
- 19. Migration
 - Definition
 - Migrationsprozesse
 - Methoden und Techniken der Software Migration
 - Fallstudien

Literatur:

- Harry Sneed: "Softwaremigration in der Praxis: Übertragung alter Softwaresysteme in eine moderne Umgebung (Wirtschaftsinformatik)", dpunkt.verlag
- Daniel Liebhart, Guido Schmutz, Marcel Lattmann, Markus Heinisch, Michael Könings, Mischa Kölliker, Perry Pakull, Peter Welkenbach : "Integration Architecture Blueprint: Leitfaden zur Konstruktion Integrationslösungen", Carl Hanser Verlag GmbH & Co. KG
- Gregor Hohpe, Bobby Woolf: "Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions", Addison Wesley
- Stefan Conrad, Wilhelm Hasselbring, Arne Koschel, Roland Tritsch: "Enterprise Application Integration: Grundlagen - Konzepte - Entwurfsmuster - Praxisbeispiele"

Implementierung von Informationssystemen			
Modulkürzel:	IM_IIS	SPO-Nr.:	6.2.5
Zuordnung zum Curriculum:	Studiengang u. -richtung	Art des Moduls	Studiensemester
	Informatik - Master	Pflichtfach	1
Modulverantwortliche(r):	Stiehl, Volker		
Sprache:	Deutsch		
Leistungspunkte / SWS:	5 ECTS / 4 SWS		
Arbeitsaufwand:	Kontaktstunden:	47 h	
	Selbststudium:	78 h	
	Gesamtaufwand:	125 h	
Lehrveranstaltungen des Moduls:	6.2.5 Implementierung von Informationssystemen (IM_IIS)		
Lehrformen des Moduls:	IM_IIS: SU/Ü - seminaristischer Unterricht/Übung		
Prüfungsleistungen:	6.2.5 prA - praktische Arbeit/Studienarbeit		
	<ul style="list-style-type: none">Entwicklungsumgebung(en), Software-Werkzeuge, -Frameworks und Programmiersprachen werden zu Beginn der Vorlesung bekanntgegeben und sind für die Teilnehmer verpflichtend.Einzelne Themengebiete werden von Teilnehmern einzeln oder in Kleingruppen vorbereitet und ggf. im Plenum in Kurzvorträgen vorgestellt.		
Voraussetzungen gemäß SPO:			
Keine			
Empfohlene Voraussetzungen:			
Fortgeschrittene Java-Kenntnisse; grundlegende Kenntnisse über betriebswirtschaftliche Anwendungssysteme und Geschäftsprozesse			
Angestrebte Lernergebnisse:			
Nach dem Besuch der Lehrveranstaltung sind die Studierenden in der Lage, <ul style="list-style-type: none">wesentliche Denkansätze und Konzepte für verteilte, mehrschichtige Anwendungssysteme zu benennen und diese im Detail zu erklärenMethoden des Software-Engineerings für verteilte, mehrschichtige Anwendungssysteme zu beschreiben und sie effektiv anzuwendeneine professionelle Arbeitsumgebung für die Softwareentwicklung aufzusetzen und einen gemeinsamen Code unterschiedlichster Sprachen in einem Versionierungssystem zu verwaltenSoftware-Architekturen für verteilte, mehrschichtige Anwendungssysteme zu entwerfen, zu bewerten und diese einzusetzenPersistenzmechanismen für verschiedene Anwendungsfälle der Datenhaltung einzuschätzen und diese in einem konkreten Projektkontext zu implementierendie Bedeutung prozessgesteuerter Anwendungssysteme zu interpretieren und Prozesse auf Basis BPMN zu implementierenverschiedenste Möglichkeiten verteilter Methodenaufrufe und Serviceimplementierungstechnologien zu beschreiben, sie in konkreten Projektsituationen anzuwenden und deren Einsatz in der jeweiligen Projektsituation zu beurteilenunterschiedlichste Datenaustauschformate zwischen Client und Server zu erklären, diese mit den jeweiligen Client- bzw. Servertechnologien zu verarbeiten und deren Einsatz für verschiedenste Anwendungsfälle zu bewerten			

- die Bedeutung von Messaging/Integration/Enterprise Integration Patterns in prozessgesteuerten Anwendungssystemen zu erläutern und sie zur Lösung von Integrationsproblemen zielgerichtet anzuwenden
- während der gesamten Veranstaltungsdauer unter freier Zeiteinteilung in Teams an der Lösung eines selbstgewählten komplexen fachlichen Problems zu arbeiten und Arbeitsergebnisse zu präsentieren

Inhalt:

- Professionelle Arbeitsumgebung in der Software-Entwicklung, einschl. Quellcode-Versionsverwaltung
- Software-Architekturen verteilter, mehrschichtiger Informationssysteme
- Methodik zur Erstellung prozessgesteuerter Anwendungssysteme
- Prozessimplementierungen auf Basis BPMN
- Datenaustauschformate zwischen Client und Server: XML vs. JSON
- Verteilte Methodenaufrufe und Serviceimplementierungstechnologien (SOAP-Webservices vs. REST-Webservices)
- Persistenzmechanismen für verschiedene Anwendungsfälle der Datenhaltung (JDBC vs. JPA)
- Messaging, Integration und Enterprise Integration Patterns

Literatur:

- STIEHL, Volker, 2013. *Prozessgesteuerte Anwendungen entwickeln und ausführen mit BPMN: wie flexible Anwendungsarchitekturen wirklich erreicht werden können*. 1. Auflage. Heidelberg: dpunkt-Verl.. ISBN 978-3-86490-007-5, 3-86490-007-7
- FREUND, Jakob und Bernd RÜCKER, 2017. *Praxishandbuch BPMN: mit Einführung in CMMN und DMN*. 5. Auflage. München: Hanser. ISBN 978-3-446-45054-7, 3-446-45054-8
- SILVER, Bruce, 2016. *DMN method and style: the practitioner's guide to decision modeling with business rules*. Altadena, CA: Cody-Cassidy Press. ISBN 978-0-9823681-5-2
- SILVER, Bruce, 2012. *BPMN, Methode und Stil: mit dem BPMN Handbuch für die Prozessautomatisierung*. 2. Auflage. Aptos, Calif.: Cody-Cassidy Press. ISBN 978-0-9823681-2-1, 0-9823681-2-7