



Empfehlung für Studierende zur Gestaltung des Studienprogramms

Pflichtbereich

Name	LP	Semester
Pflichtveranstaltung Formale Methoden	6	1
Pflichtveranstaltung Software Engineering	6	1
Pflichtveranstaltung Datenbanken	6	1
Pflichtveranstaltung Human-Computer-Interaction	6	2
Pflichtveranstaltung Verteilte Systeme	6	2
Gesamt-LP Pflichtbereich	30	

Die Pflichtveranstaltungen müssen erfolgreich besucht werden und tragen auf jeden Fall zur Abschlussnote bei. **Die Noten in den Pflichtveranstaltungen können nicht durch Noten in anderen Modulen ersetzt werden!**

Wahlbereich

1) Projektmanagement:

Auf Grund der Bedeutung des Themas wird allen Studierenden empfohlen, nach der Vorlesungszeit des ersten Semesters die Wahlveranstaltung Projektmanagement aus dem Software Engineering-Bereich zu besuchen:

Name	LP	Semester
Projektmanagement	6	1

Die Veranstaltung Projektmanagement deckt die laut Prüfungsordnung mindestens zu erbringenden 6 LP im Bereich Software Engineering ab.

2) Ausgewogene Vorlesungsauswahl:

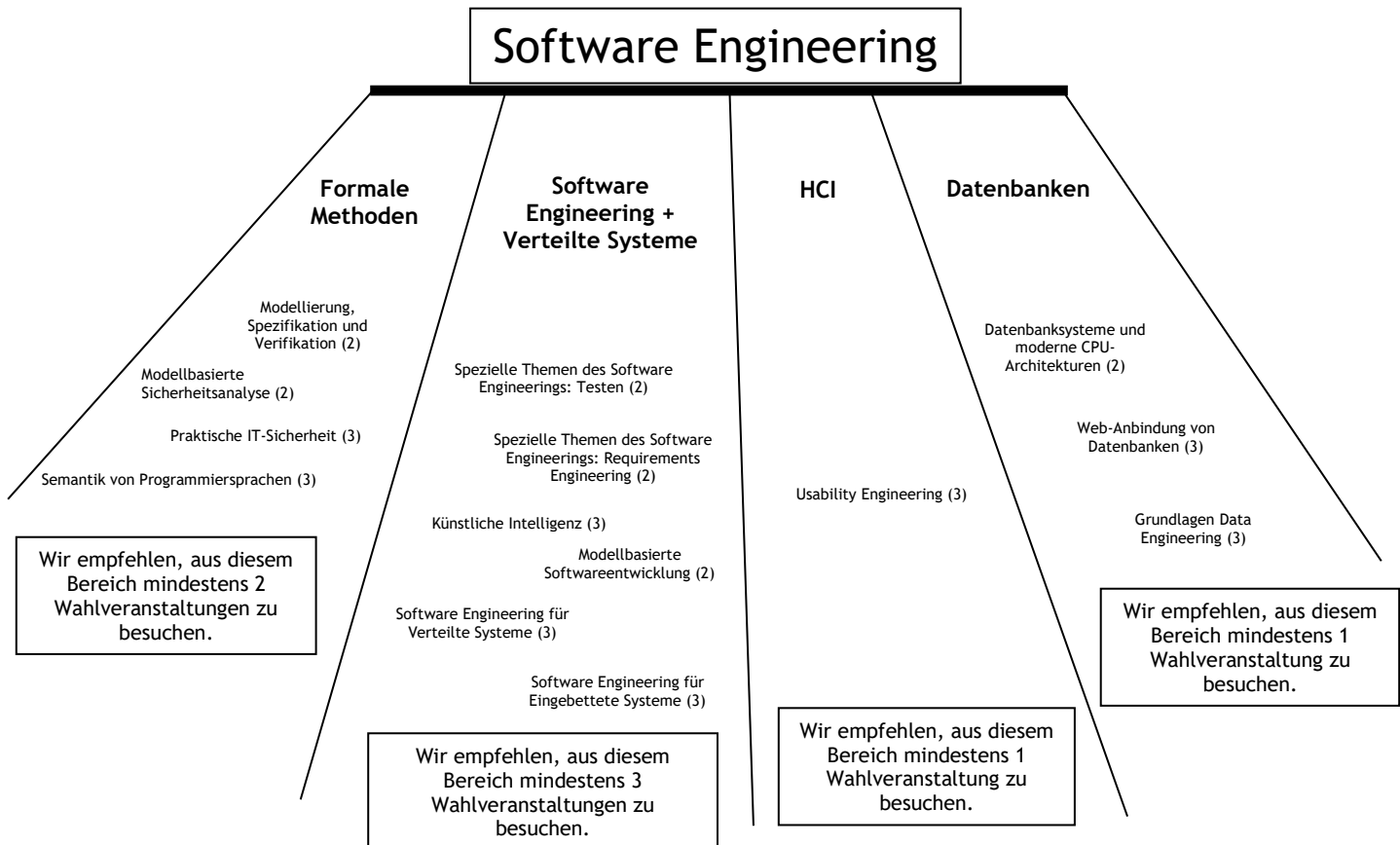
Aus dem restlichen Wahlbereich sind dann weitere 34 LP zu erbringen, wobei genau ein Seminar belegt werden muss und kann und mindestens 6 LP Lehrveranstaltungen im Bereich Formale Methoden gewählt werden müssen.



SOFTWARE ENGINEERING

Elite Graduate Program

Die folgende Grafik gibt einen Überblick über die weiteren Wahlveranstaltungen in den fachlichen Bereichen des Studiengangs. In Klammern ist jeweils das Semester angegeben, für das die Vorlesung empfohlen wird (bzw. in der sie angeboten wird). Unsere Empfehlung ist in den Rechtecken angegeben



3) Seminare und Projekte

Jeder Studierende des Elitestudiengangs sollte während seines Studiums an einem Spezialprojekt („Pushing-the-limits-Projekt“) teilnehmen. Beispiele für solche Projekte der bisherigen Jahrgänge sind XENAKIS, Sightplayer, Monopedia und CATS. Die Projekte gehen dabei immer auf studentische Initiativen und Ideen zurück. Es gibt damit für die Studierenden die Möglichkeit ein interessantes und herausforderndes Projekt in Form einer Lehrveranstaltung zu bearbeiten. Die Spezialprojekte beginnen im 2. Semester (Ideenfindung und Konzeption) und dauern bis in das 3. Semester. Die Pushing-the-limits-Projekte werden als Seminare abgerechnet, womit auch der Forderung der Prüfungsordnung genüge getan ist, genau ein Seminar zu absolvieren.

Reguläre Seminare (Ausarbeitung und Vortrag) werden nur im 3. Semester angeboten. Üblicherweise findet höchstens ein Seminar in jedem der fünf Bereiche des Elitestudiengangs statt. Es ist daher zu empfehlen, an einem Pushing-the-limits-Projekt teilzunehmen, um das nötige Seminar sicher abgehandelt zu haben.



4) Ringvorlesung/Erweitertes Angebot

Neben den Lehrveranstaltungen, die für die Erreichung der Leistungspunkte im Wahlbereich zählen, empfehlen wir auch die Teilnahme an weiterbildenden Veranstaltungen (z.B. Ringvorlesung) aus dem aktuellen Angebot entsprechend der eigenen Interessenslage. Insbesondere die Teilnahme an der eigens organisierten Ringvorlesung wird erwartet.

Softskill-Bereich

Der Studiengang bietet die folgenden Soft-Skill-Veranstaltungen an:

- „Präsentationstechnik“
- „Schreibtraining“
- „Erfolg im Team“
- „Führungskompetenz“
- „Moderationstechnik“
- „Change-Prozesse in verschiedenen Arbeitskontexten“
- „Interkulturelle Kompetenz“

Diese müssen alle Studierenden des Elitestudiengangs besuchen. Die Veranstaltungen entsprechen den laut Prüfungsordnung erforderlichen **10 LP** im Soft-Skill-Bereich. Die Soft-Skill-Veranstaltungen finden in den vorlesungsfreien Zeiten zwischen erstem und zweitem und drittem und viertem Semester sowie an einem verlängerten Wochenende im zweiten Semester statt.

Industriepraktikum

Der Studiengang verlangt die Ableistung eines Industriepraktikums von mindestens 8 Wochen Dauer (Vollzeit). Das Industriepraktikum sollte bei einem der Industriepartner des Studiengangs absolviert werden, wofür die Partnerunternehmen geeignete Angebote machen. Das Praktikum zählt **10 LP**.

Vorbereitung und Anmeldung

Mit der Planung des Praktikums, das in der vorlesungsfreien Zeit zwischen zweitem und drittem Semester stattfindet, sollte zu Beginn des zweiten Semesters begonnen werden. Alle Studierenden erhalten vom Koordinator des Studiengangs Informationen zu den Praktikumsangeboten der Partnerunternehmen. Die Bewerbung für die Praktikumsplätze und die Auswahl der Praktikanten durch die Firmenpartner erfolgt im Sommersemester. Praktika müssen mit dem Studiengang abgestimmt werden und mit den Leitlinien konform sein. Insbesondere sollten die Praktika zum Vorlesungsbeginn des Wintersemesters beendet sein.

Masterarbeit

Die Masterarbeit wird im Allgemeinen im **vierten Semester** erstellt und sie ist mit **30 LP** bewertet. Die Masterarbeit kann an einem der beteiligten Lehrstühle oder bei einem Firmenpartner des Elitestudiengangs erstellt werden. Die Erstellung der Masterarbeit bietet die Möglichkeit zu einem Auslandsaufenthalt.

Vorbereitung und Anmeldung

Zur Vorbereitung der Masterarbeit sollte bis spätestens Mitte des dritten Semesters Kontakt mit dem Koordinator des Studiengangs hergestellt werden, damit mit der Vorbereitung begonnen werden kann. Für **einen Auslandsaufenthalt in Japan** im Rahmen der Kooperation des Elitestudiengangs mit dem **National Institute of**



SOFTWARE ENGINEERING

Elite Graduate Program

Informatics (NII) in Tokio ist eine **Bewerbung bereits im 2. Semester** (Sommersemester) empfehlenswert. Bei Interesse wenden Sie sich frühzeitig im 2. Semester an den Koordinator.

Wichtige Dokumente zum Studium

Sie finden wichtige Dokumente und Formulare zum Studiengang im Internet:

- Prüfungsordnung: <https://www.uni-augsburg.de/de/services/rechtssammlung/fai-elite-software-engineering/>
- Modulhandbuch: <https://mhb.uni-augsburg.de/MasterStudiengaenge/Master+of+Science/Software+Engineering+%28Hauptfach%29/POVersion+2013/>
- Studienempfehlung: <https://elite-se.informatik.uni-augsburg.de/downloads/>
- Formulare des Prüfungsamtes (Antrag Masterarbeit, Antrag Zeugniserstellung): <https://www.uni-augsburg.de/de/studium/organisation-beratung/pruefungen/infos-und-antrage/fakultat-fur-angewandte-informatik-pruefungen/>
- Hinweise zur Masterarbeit: <https://elite-se.informatik.uni-augsburg.de/wp-content/uploads/2017/10/Hinweise-Masterarbeit-und-Auslandsaufenthalt.pdf>

Abschluss des Studiums und Zeugnis

Bitte beachten Sie, dass zum Ende des Studiums die Urkunde/das Zeugnis nicht automatisch nach der Abgabe der Masterarbeit ausgestellt werden. Die Zeugniserstellung (und damit das Ende des Studiums) muss beim Prüfungsamt beantragt werden, da das Prüfungsamt nicht wissen kann, ob Sie nicht noch weitere Prüfungsleistungen nach der Masterarbeit erbringen wollen.



Organisatorisches

Kommunikation

Wichtige Informationen zum Studienverlauf, Termine u.ä. werden über die Mailingliste des Jahrgangs verbreitet. Daher ist es sehr wichtig, als Student über diesen Verteiler erreichbar zu sein. Sollte sich Ihre E-Mail-Adresse ändern, informieren Sie bitte den Koordinator, damit Sie nicht von wichtigen Informationen abgeschnitten werden.

Informationen der Geschäftsstelle des Elitenetzwerks Bayern werden über die im ENB-Intranet hinterlegte E-Mail-Adresse verschickt. Auch dort sollte daher immer eine aktuelle E-Mail-Adresse eingetragen sein. Dies gilt über das Ende des Studiums hinaus, um an Alumni-Aktivitäten teilnehmen zu können

Befragungen

Das Elitenetzwerk führt regelmäßig Befragungen der Studierenden und der Alumni durch. Die Teilnahme an der Befragung ist aus verschiedenen Gründen wichtig, erstens braucht das ENB insgesamt hohe Zahlen, um zu dokumentieren, dass die Angebote des Elitenetzwerks entsprechenden Wertschätzung erfahren, womit sich dann auch die Ausgaben für das Elitenetzwerk begründen lassen. Außerdem ist eine hohe Beteiligung von uns wichtig, um dem ENB zu zeigen, dass unser Studiengang sich in das ENB einbringt. Das kann wichtig sein, wenn das ENB bewertet, ob unser Studiengang weiterlaufen kann. Nehmen Sie daher die Möglichkeit wahr, an den Befragungen teilzunehmen.



Lehrveranstaltungen im Elitestudiengang Software Engineering

Die aktuellsten Informationen zu den Modulen finden sich immer im Modulhandbuch des Studiengangs, das auf der Webseite des Prüfungsamts verlinkt ist:

<https://mhb.uni->

[augsburg.de/MasterStudiengaenge/Master+of+Science/Software+Engineering+%28Hauptfach%29/POVersion+2013/](https://mhb.uni-augsburg.de/MasterStudiengaenge/Master+of+Science/Software+Engineering+%28Hauptfach%29/POVersion+2013/)

Bereich Software Engineering

- **Pflichtveranstaltung Software Engineering**

Die Vorlesung gibt einen Überblick über Methoden zur systematischen Entwicklung von Software. Ausgehend von einer Diskussion des Softwarelebenszyklus behandelt die Vorlesung insbesondere die Phasen der Softwareentwicklung: Analyse, Spezifikation, Design, Implementierung, Wartung. Diese spiegeln sich in den behandelten Softwareprozessen wieder, von denen der Unified Process den größten Raum einnimmt. In Analyse und Design sind Modularisierung, Abstraktion und Objekt-Orientierung wesentliche Themen, ebenso wie der Einsatz der UML. Ein weiteres wichtiges Thema sind Design Patterns. Aus der Implementierungsphase wird insbesondere die Qualitätssicherung durch Testen und Reviews thematisiert.

LP: 6

- **Projektmanagement**

Der Erfolg eines Softwareentwicklungsprojekts hängt wesentlich von der Güte des Projektmanagements ab. Wesentliche Ziele des Projektmanagements bestehen darin, die Produktivität zu erhöhen, die Qualität sicherzustellen und vorgegebene Kosten- und Zeitrahmen einzuhalten. In dieser Vorlesung werden die wesentlichen Aufgaben, Prozesse, Methoden und Werkzeuge des Projektmanagement vorgestellt und an praktischen Beispielen eingeübt.

Insbesondere werden die folgenden Themen behandelt:

- Softwaretechnik und Projektmanagement
- Projektauftrag und Projektinitialisierung
- Projektstrukturen, Prozessmodelle und Personalaktivitäten
- Projektplanung und Schätzverfahren
- Projektsteuerung und -Kontrolle
- Qualitätsmanagement
- Risikomanagement
- Kommunikation und Teamführung
- Projektabschluss und Prozessverbesserung

LP: 6



SOFTWARE ENGINEERING

Elite Graduate Program

- **Software Engineering für Embedded Systems**
Eingebettete Systeme sind für einen spezifischen Zweck in einer technischen Umgebung entworfene, eingebaute und betriebene informationsverarbeitende Systeme, wie sie etwa in Automobilen, Flugzeugen, Haushaltsgeräten, Unterhaltungselektronik oder Mobilfunkgeräten zu finden sind. Prägende Merkmale sind die Umgebungskommunikation durch Sensoren und Aktuatoren und Realzeitabhängigkeiten. Die Vorlesung gibt einen Überblick in den Entwurf, die Implementierung und das technische Umfeld eingebetteter Systeme.

Schlagwörter: synchrone und asynchrone Modellierung; Spezifikationsprachen (z. B. Statecharts, Esterel, VHDL, SystemC, Giotto), gezeitete Automaten, hybride Automaten; eingebettete Hardware, Sensoren, Aktuatoren, Application-Specific Circuits; eingebettete Betriebssysteme, Scheduling, Middleware; Hardware/Software Co-Design

LP: 6

- **Selected Topics in Software Engineering: Testen**
Testen dient der Bewertung und Verbesserung der Produktqualität durch die Identifikation von Mängeln und Problemen. Der Softwaretest konzentriert sich dabei auf die dynamische Überprüfung des Softwareverhaltens gegenüber einem erwarteten Verhalten basierend auf einer endlichen Menge von Testfällen, die geeignet aus dem im allgemeinen unendlichen Ablaufraum der Software auszuwählen sind. Die Vorlesung stellt die Grundlagen des Softwaretests vor, erläutert die Einbindung des Softwaretests in den Softwareentwicklungsprozess und beschreibt und diskutiert Testmethoden und -werkzeuge sowie Verfahren des Testmanagements.

Schlagwörter: Software-Qualität, Testprozess, funktionales Testen, strukturelles Testen, diversifizierendes Testen, modellbasiertes Testen, Testen eingebetteter Software (Konformanztesten), Testen objektorientierter Software, Integrationstest, Testausführung

LP: 5

- **Selected Topics in Software Engineering: Requirements Engineering**
Requirements Engineering (RE) ist eine entscheidende Aufgabe/Phase jedes Entwicklungsprojektes. Hierbei zu erarbeitende Anforderungs- und Systemspezifikationen sind Grundlage für die weitere Entwicklung, Integration und Abnahme des zu entwickelnden Systems. Sie bilden die Abstimmungs- und Vertragsgrundlage für alle Projektbeteiligten (Kunden, Nutzer, Entwickler u.a.) und sind Ausgangspunkt für Aufwandschätzung und Planung des Projektes.

Die Vorlesung beginnt mit einer Einführung in die Ziele, Aufgaben und Inhalte des Requirements Engineering. Nach einem Überblick über Anforderungsarten, wesentliche Prozessschritte, Methoden und Techniken der Anforderungsentwicklung und Spezifikation werden folgende Themen des RE vertiefend behandelt: Anforderungserhebung und ihre strukturierte Dokumentation, Use Case/Szenario-Modellierung, Nicht-funktionale



SOFTWARE ENGINEERING

Elite Graduate Program

Anforderungen, Requirements Management, Systemmodelle in RE und Nutzerschnittstellen.

Diese Themen und ihre RE-Techniken werden anhand von Fallstudien aus Forschung und industrieller Praxis vermittelt und in praktischen Übungen vertiefend bearbeitet.

LP: 6

- **Modellbasierte Softwareentwicklung**

Die modellgetriebene Softwareentwicklung setzt sich das Ziel, Softwaresysteme möglichst weitgehend durch (semi-)formale Modelle zu beschreiben und aus diesen Modellen möglichst viele Artefakte von Softwaresystemen generativ ableiten zu können. In diese Vorgehensweise reihen sich der "Model-Driven Architecture"-Ansatz der OMG, der "Software Factory"-Ansatz von Microsoft, Modellinterprete wie die "Executable UML" und genereller die Verwendung von "Domain-Specific Languages" und das generative Programmieren ein. Die Vorlesung gibt einen Überblick über das Gebiet der modellgetriebenen Softwareentwicklung anhand der Themenschwerpunkte: Modellierung, Metamodellierung, Transformationen und Codegenerierung.

Schlagwörter: Modellgetriebene Entwicklungsansätze (z. B. Model-Driven Architecture), Modellierungssprachen (z. B. Unified Modeling Language, Object Constraint Language), Metamodellierung (z. B. Meta-Object Facility, Eclipse Modeling Framework), Modellaustausch (z. B. XML Metadata Interchange), Model-to-Model- und Model-to-Text-Transformationen (z. B. Query/View/Transformation, Graphtransformationen, Java Emitter Templates, xPand), Modelltransformationwerkzeuge (z. B. openArchitectureWare, AndroMDA)

LP: 5

- **Künstliche Intelligenz**

Die Vorlesung grenzt zunächst die (klassische) künstliche Intelligenz von Verfahren des maschinellen Lernens ab und erläutert ihre Rolle in industriellen Anwendungen (Industrie 4.0, Smart Factory, Smart Grids). Es wird auf unterschiedliche Methoden und Modelle wie Entscheidungsbäume und -wälder, Ensemble-Methoden, Regression, numerische, kontinuierliche Optimierung (Gradientenverfahren, SGD, ADAM) auch unter Nebenbedingungen, probabilistische Modelle, Gauss-Prozesse, aktives Lernen, neuronale Netze, Faltungsnetze (CNN) und rekurrente neuronale Netze eingegangen. Darüber hinaus werden Themen des Natural Language Processing bzw. Verarbeitung von diskreten Daten diskutiert.

LP: 5



Bereich Formale Methoden und IT-Sicherheit

- **Pflichtveranstaltung Formale Methoden**

Die Entwicklung beweisbar korrekter Software ist eines der wichtigsten Ziele des Software-Engineering. Für einen Korrektheitsnachweis sind einerseits die Anforderungen, gemäß denen sich die Software verhalten soll, formal zu spezifizieren und andererseits das Verhalten der Software in einem adäquaten semantischen Modell zu erfassen. Die Vorlesung legt die mathematischen Grundlagen für die Spezifikation und die Semantik von sequentiellen Programmen, stellt Beweismethoden und ihre Umsetzung in einem Theorembeweiser vor und gibt eine Einführung in die methodische Entwicklung korrekter sequentieller Programme.

Schlagwörter: Aussagenlogik, Prädikatenlogik, Beweiskalküle, Induktion, algebraische Spezifikationen, operationale Semantik, Hoare-Logik, dynamische Logik, abstrakte Datentypen, Verfeinerung; Theorembeweiser (z. B. KIV)
LP: 6

- **Modellierung, Spezifikation und Verifikation**

Reaktive Systeme verarbeiten Informationen unter Reaktion auf und in Interaktion mit einer Umgebung. Typische Vertreter sind Betriebssysteme und Software für Steuergeräte; häufig finden sich reaktive Systeme in sicherheitskritischen Bereichen, etwa medizinischen Anwendungen, Zahlungssystemen oder Zugangskontrollsystemen, in denen hohe Zuverlässigkeitsanforderungen gestellt werden. Die prägenden Eigenschaften reaktiver Systeme sind Nebenläufigkeit, Kommunikation und Nichtterminierung. Die Vorlesung stellt Ansätze zur Modellierung und Spezifikation reaktiver Systeme vor und gibt eine Einführung in Verfeinerungs- und Verifikationstechniken für zuverlässige, reaktive Systeme.

Schlagwörter: Transitionssysteme, Sicherheits- und Lebendigkeitseigenschaften; strombasierte Spezifikationen; nebenläufige Termersetzungssysteme (z. B. Maude); Prozeßalgebren (z. B. Calculus of Communicating Systems, Communicating Sequential Processes, pi-Kalkül), Simulation und Bisimulation; Temporallogik, symbolisches Model checking, automatenbasiertes Model checking, Abstraktion; Temporal Logic of Actions, Verfeinerung
LP: 6

- **Modellbasierte Sicherheitsanalyse**

Am 26 April 1986 wurde offensichtlich, wie kritisch eingebettete Systeme sein können. An diesem Tag kam es im Block 4 des Kernkraftwerks Tschernobyl zu einer unkontrollierten Kernschmelze. Folgen davon waren Tausende Tote, mehrere Hundert Quadratkilometer kontaminierten Bodens und radioaktiver Fallout in einem Großteil von Zentraleuropa.

Das Ziel dieser Lehrveranstaltung ist es, in den State-of-the-art bei Sicherheitsanalysetechniken einzuführen. Diese Techniken zielen auf die Analyse



SOFTWARE ENGINEERING

Elite Graduate Program

sicherheitskritischer Systeme und liefern Designrichtlinien, um Systeme sicherer zu machen. Der Hauptfokus der Lehrveranstaltung liegt auf der ständig wachsenden Bedeutung von Software in kritischen, eingebetteten Systemen. Nur das korrekte Zusammenspiel zwischen der Hardware, der Steuerungssoftware und den Umgebungsbedingungen garantiert Sicherheit. Die Lehrveranstaltung gibt eine Einführung in traditionelle Ansätze in der Sicherheitsanalyse, ebenso wie in Leading-Edge-Techniken, die auf formalen Methoden basieren. Ebenfalls behandelt werden die Grundlagen von quantitativen und probabilistischen Bewertungsmethoden und die Notwendigkeiten einer Zertifizierung.

LP: 6

- **Praktische IT-Sicherheit**

Diese Vorlesung gibt eine Einführung in grundlegende Konzepte, Techniken, Methoden und Anwendungen der IT-Sicherheit. Die Vorlesung behandelt dabei folgende Themen:

- Die wichtigsten Bedrohungen und Angriffe, Grundbegriffe der IT-Sicherheit und Ziele der Disziplin wie Integrität, Vertraulichkeit, Verfügbarkeit und Authentizität.
- Konzepte der Kryptographie sowie ausgewählte kryptographische Verschlüsselungs-, Signier- und Hashverfahren.
- Zugriffskontrollsysteme, das heißt, Mechanismen um den Zugriff auf sensible Daten zu überwachen und zu kontrollieren.
- Konzepte und Verfahren der Authentifikation (d.h. der Überprüfung einer behaupteten Identität). Dabei werden insbesondere Herausforderungen der aktuellen biometrischen Technologien diskutiert.
- Techniken (TLS, IPSEC) und Technologien (VPN, Firewalls) der Netzwerksicherheit.
- Security Engineering, das heißt, die systematische Vorgehensweise zur Konstruktion sicherer IT-Systeme, sowie dabei eingesetzte Methoden, werden vorgestellt. Darüber hinaus werden die dabei zu berücksichtigen rechtliche Aspekte angesprochen.

LP: 3

- **Semantik von Programmiersprachen**

Es werden drei klassische Ansätze zur Semantik von Programmiersprachen besprochen: In der operationalen Semantik wird die Bedeutung eines programmiersprachlichen Konstrukts dadurch festgelegt, wie es auszuführen ist und zu welchen Effekten es führt. In der denotationellen Semantik wird von der spezifischen Ausführungsart abstrahiert und lediglich der Effekt eines Konstrukts betrachtet. Schließlich wird im axiomatischen Ansatz der Effekt eines Konstrukts durch Zusicherungen spezifiziert.

Als Beispielsprache für die unterschiedlichen semantischen Ansätze dient eine einfache, imperative WHILE-Sprache, die im Verlauf der Vorlesung um syntaktische Konstrukte erweitert wird.

LP: 5



Bereich Datenbanken

- **Pflichtveranstaltung Datenbanken**

Die Vorlesung vermittelt Kenntnisse die zum Entwurf und Betrieb von Datenbanksystemen benötigt werden. Behandelt wird das relationale Datenmodell das modernen Datenbanken zu Grunde liegt, Modellierung- und Entwurfstechniken für das Datenbankdesign und relationale Anfragesprachen, sowie deren Grundlage, die relationale Algebra. Außerdem behandelt wird die Anbindung von Datenbanken an moderne Programmiersprachen wie JAVA und Datenbanke Erweiterungen welche die Verarbeitung von XML Daten oder objektorientierte Datenmodelle unterstützen. Weiterer Inhalt der Vorlesung sind die Technologien, die bei der Implementierung von Datenbanksystemen eingesetzt werden, z.B. zur Anfragebearbeitung, Transaktionsverwaltung, Mehrbenutzersynchronisation, Fehlerbehandlung, etc.

LP: 6
- **Web-Datenbanken**

In vielen aktuellen Einsatzgebieten sind Datenbanken über Web-basierte Schnittstellen für Benutzer zugänglich, sei es beim Einkauf bei einem Internethändler, bei der Kundenbetreuung, beim Online-Banking, etc. Die große Zahl an gleichzeitigen Benutzern und die Verteilung der Anwendungen auf mehrere verschiedene Hard- und Softwaresysteme wie Web-Server, Datenbank-Server, Application-Server, etc haben zur Entwicklung spezieller Softwarearchitekturen und Entwicklungsparadigmen geführt. Die Vorlesung behandelt aktuelle Programmiersprachen, Technologien und Architekturen für moderne Web-basierte Informationssysteme. Diese Veranstaltung soll neben den theoretischen Grundlagen auch praktische Erfahrung beim Umgang mit den relevanten Technologien vermitteln. Aufgabe der Studenten ist deshalb neben dem Erlernen der theoretischen Grundlagen auch der Entwurf und die Implementierung einer modernen Web-basierten Datenbankanwendung.

LP: 5
- **Datenbanksysteme und moderne CPU-Architekturen**

Understand the interaction between database systems resp. algorithms and modern computer architecture (esp. CPU, Cache, Primary Storage) and learn how to develop resp. to modify the internals of database systems in order to make use of the properties of these computer architecture features.

LP: 6
- **Grundlagen Data Engineering**

Die Veranstaltung behandelt grundlegende Techniken zur Organisation und effizienten Analyse (sehr) großer Datenmengen. Dazu werden Verfahren der verteilten Datenhaltung in Abhängigkeit von der Systemarchitektur (z.B. Mehrkernsysteme, Multiprozessorsysteme, Cluster), verteilte und parallele Datenanalyse, Statistik-gestützte Optimierung von Analyseprozessen, map/reduce Techniken und ihre Verallgemeinerung zur verteilten



SOFTWARE ENGINEERING

Elite Graduate Program

Anfrageverarbeitung vorgestellt. Darüber hinaus werden Anwendungen in Systemen wie Hadoop und in NoSQL Systemen, die gesellschaftlichen Auswirkungen der Datenanalyse und die Bedeutung des Datenschutzes thematisiert.

LP: 6

Bereich Verteilte Systeme

- **Pflichtveranstaltung Verteilte Systeme**

Inhalte dieser Veranstaltung sind grundlegende Begriffe und Techniken der Entwicklung verteilter Systeme:

1. Introduction
2. Communication
 - Fundamental Interaction Model
 - Networks
 - Protocol and Services
 - Interprocess Communication
3. Processes, Synchronization and Coordination
 - Processes and Threads
 - Synchronization
 - Coordination
4. Service-Oriented Architecture und Web Services
 - SOA
 - Web Service
 - REST
 - Web Service Composition

LP: 6

- **Software Engineering für Verteilte Systeme**

Diese Veranstaltung beschäftigt sich mit Softwarearchitekturen verteilter System, mit semantischen Techniken und mit Multi-Agenten-Systemen:

1. Introduction
2. Software Architectures
 - Software Architectures and Organizations
 - Design of Software Architectures
 - Documentation of Software Architectures
 - Evaluation of Software Architectures
3. Semantic Technologies
 - Introduction to the Semantic Web
 - Ontology Languages for the Semantic Web
 - Description Logics
 - Reasoning with OWL
 - Description Logic Reasoning
 - Advanced Reasoning Techniques
 - Summary



SOFTWARE ENGINEERING

Elite Graduate Program

4. Multi-Agent Systems

- Intelligent Agents
- Agent architectures
- Reactive and hybrid agents
- Heuristic approach for decision finding
- Interaction in Multi-Agent systems and negotiation
- Coordination and Cooperation through communication

LP: 6

Bereich Human-Computer-Interaction

- **Pflichtveranstaltung HCI**

Die Entwicklung multipler Medien zur Informationsdarbietung und zur Gestaltung der Mensch-Maschine-Schnittstelle hat in nur wenigen Jahren den Umgang mit Computern grundlegend verändert und wesentlich dazu beigetragen, Computertechnologie einer breiten Benutzerschicht zugänglich zu machen. In der Vorlesung werden zunächst die theoretischen Grundlagen moderner Interaktionsparadigmen vorgestellt bevor auf spezifische Verfahren zur Erkennung und Interpretation von Benutzereingaben sowie zur Generierung und Synchronisation von Systemausgaben eingegangen wird. Die hohe Komplexität der verwendeten Ein- und Ausgabemodalitäten erfordert zudem die Berücksichtigung spezifischer Usabilityrichtlinien beim Entwurf interaktiver Systeme, die im weiteren Verlauf der Vorlesung eingeführt werden.

LP: 6

- **Usability Engineering**

Durch die zunehmende Verbreitung von Computern in allen Lebensreichen und deren Einbettung in die natürliche Umgebung des Benutzers wird die Gestaltung der Mensch-Maschine Interaktion zu einer großen Herausforderung. Ziel der Veranstaltung ist es grundlegende Techniken zum Entwurf von Benutzungsoberflächen zu vermitteln. Während bislang rein technische Aspekte dominierten, geht der Trend hin zu Entwurfsprozessen, die den Nutzer in den Vordergrund stellen und ihn bereits in der Entwurfsphase miteinbeziehen. Behandelt werden folgende Themen:

- Entwurfsprinzipien und Normen
- Menschliche Informations-, Handlungs- und Wahrnehmungsprozesse
- Benutzerzentrierter Entwicklungsprozess
- Evaluierung von interaktiven Systemen

Zu Beginn der Veranstaltung werden Grundlagen des Usability Engineering erarbeitet. Die erlernten Techniken werden dann anhand eines konkreten Projekts umgesetzt, wobei alle Phasen eines benutzerzentrierten Entwicklungsprozesses zumindest ansatzweise durchlaufen werden. In Gruppen soll eine Applikation analysiert, entworfen, implementiert und evaluiert werden.

LP: 6



Bereich Soft Skills

- „Präsentationstechnik“
- „Schreibtraining“
- „Erfolg im Team“
- „Führungskompetenz“
- „Moderationstechnik“
- „Change-Prozesse in verschiedenen Arbeitskontexten“
- „Interkulturelle Kompetenz“