

Modulhandbuch  
Elite-Masterstudiengang  
Software Engineering  
**Sommersemester 2012**



<b>1. Modultitel</b>	Mensch-Maschine Interaktion
<b>2. Englischer Modultitel</b>	Human-Computer Interaction
<b>3. Modulbereich</b>	<input checked="" type="checkbox"/> Pflichtbereich <input type="checkbox"/> Wahlbereich
<b>4. Fachgebiet</b>	Human-Computer Interaction
<b>5. Modulbeauftragte/r</b>	Prof. Dr. Elisabeth Andre
<b>6. Lernziele/Lernergebnis</b> (allgemein für das Modul)	Die Studierenden sollen grundlegende Kenntnisse zu Konzepten, Methoden und Techniken zur Realisierung multimedialer Benutzungsoberflächen erwerben. Sie sollen in die Lage versetzt werden, die erworbenen Kenntnisse im Hinblick auf gegebene Anwendungsszenarien einschätzen und anwenden zu können.
<b>7. Inhalte</b> (allgemein für das Modul)	Die Veranstaltung vermittelt technische Grundlagen zur Realisierung multimedialer Benutzungsoberflächen. Behandelt werden folgende Themen: <ul style="list-style-type: none"> <li>• Interaktionsmodalitäten und -techniken</li> <li>• Medienspezifischen Analyse (Sprache, Gestik, Mimik etc)</li> <li>• Fusionsmodelle für vielkanalige Sensorsysteme (visuelle, auditive, haptische und biophysiological Sensoren)</li> <li>• Medienspezifische Generierung (Sprache, Animationen etc.)</li> <li>• Kontextsensitive Auswahl von Medien</li> <li>• Medienkoordination und –synchronisation</li> <li>• Multimedialer Zugang zu Informationssystemen und zum WWW</li> <li>• Multimodale Dialogsysteme</li> <li>• Benutzer- und Diskursmodellierung</li> <li>• Agentenbasierte Multimediale Interaktion</li> <li>• Benutzungsschnittstellen der nächsten Generation (Perzeptive Interfaces, Emotionale Interfaces, Mensch-Roboter Interaktion etc.)</li> </ul>
<b>8. Unterlagen (Skripte, Literatur)</b>	<ul style="list-style-type: none"> <li>• J. Schenk, G. Rigoll: Mensch-Maschine-Kommunikation: Grundlagen von sprach- und bildbasierten Benutzerschnittstellen</li> <li>• Daniel Jurafsky, James H. Martin: Speech and Language Processing. Pearson Prentice Hall</li> <li>• T. Mitchell: Machine Learning, McGraw Hill</li> </ul>
<b>9. Lehrform/en</b> (ggf. Anzahl der zu besuchenden LV mit Lehrform)	Vorlesung mit integrierter Übung
<b>10. Semesterempfehlung</b>	1
<b>11. Häufigkeit des Angebots</b>	Jährlich
<b>12. Inhaltliche Voraussetzungen</b>	-
<b>13. Arbeitsaufwand (gesamt)</b>	180 (Präsenz: 56h, Selbststudium: 124h)
<b>14. Anzahl der LP</b>	6
<b>15. Voraussetzungen für die Vergabe von LP/ECTS</b>	Erreichen einer Mindestpunktzahl in den Übungsaufgaben
<b>16. Prüfung</b>	Modulgesamtprüfung:Mensch-Maschine Interaktion
<b>17. Anmeldeformalitäten</b>	Anmeldung zur Prüfung in STUDIS

	Modulgesamtprüfung Mensch-Maschine Interaktion	SWS	4
<b>Summe:</b>			LP: 6

<b>1. Modultitel</b>	Requirements Engineering
<b>2. Englischer Modultitel</b>	Requirements Engineering
<b>3. Modulbereich</b>	<input type="checkbox"/> Pflichtbereich <input checked="" type="checkbox"/> Wahlbereich
<b>4. Fachgebiet</b>	Software Engineering
<b>5. Modulbeauftragte/r</b>	Prof. Dr. Dr. h.c. Manfred Broy
<b>6. Lernziele/Lernergebnis</b> (allgemein für das Modul)	<p>Studierende erlangen in diesem Modul:</p> <ul style="list-style-type: none"> <li>• Verständnis für die Bedeutung und die spezifischen Problemstellungen des Requirements Engineering</li> <li>• Einen Überblick über die Prinzipien, Techniken und Artefakte im Requirements Engineering</li> </ul>
<b>7. Inhalte</b> (allgemein für das Modul)	<p>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.</p> <p>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 Anforderungen, Requirements Management, Systemmodelle in RE und Nutzerschnittstellen.</p> <p>Diese Themen und ihre RE-Techniken werden anhand von Fallstudien aus Forschung und industrieller Praxis vermittelt und in praktischen Übungen vertiefend bearbeitet.</p>
<b>8. Unterlagen (Skripte, Literatur)</b>	<ul style="list-style-type: none"> <li>• Axel van Lamsweerde: Requirements Engineering - From System Goals to UML Models to Software Specifications</li> <li>• Sommerville, G. Kotonya: Requirements Engineering: Processes and Techniques</li> <li>• Sommerville, P. Sawyer: Requirements Engineering: A Good Practice Guide</li> <li>• K. E. Wiegers: Software Requirements</li> <li>• S. Robertson, J. Robertson: Mastering the Requirements Engineering Process</li> <li>• C. Rupp: Requirements-Engineering und -Management. Professionelle, iterative Anforderungsanalyse für die Praxis</li> <li>• Alexander, R. Stevens: Writing better requirements</li> <li>• L. Maciaszek: Requirements Analysis and System Design</li> </ul>

<b>9. Lehrform/en</b> (ggf. Anzahl der zu besuchenden LV mit Lehrform)	Vorlesung mit integrierter Übung		
<b>10. Semesterempfehlung</b>	2		
<b>11. Häufigkeit des Angebots</b>	Jährlich		
<b>12. Inhaltliche Voraussetzungen</b>	--		
<b>13. Arbeitsaufwand (gesamt)</b>	180 Stunden (Präsenz: 56h, Selbststudium: 124h)		
<b>14. Anzahl der LP</b>	6		
<b>15. Voraussetzungen für die Vergabe von LP/ECTS</b>	Schriftliche Prüfung		
<b>16. Prüfung</b>	Modulgesamtprüfung:Requirements Engineering		
<b>17. Anmeldeformalitäten</b>	Anmeldung zur Prüfung in STUDIS		
	Modulgesamtprüfung:Requirements Engineering	SWS	4
<b>Summe:</b>			LP: 6

<b>1. Modultitel</b>	Verteilte Systeme		
<b>2. Englischer Modultitel</b>	Distributed Systems		
<b>3. Modulbereich</b>	<input checked="" type="checkbox"/> Pflichtbereich	<input type="checkbox"/> Wahlbereich	
<b>4. Fachgebiet</b>	Verteilte Systeme		
<b>5. Modulbeauftragte/r</b>	Prof. Dr. Bernhard Bauer		
<b>6. Lernziele/Lernergebnis</b> (allgemein für das Modul)	Vertieftes Verständnis für verteilte Systeme. Technologien und Algorithmen für die Entwicklung verteilter Systeme verstehen, bewerten und anwenden können.		
<b>7. Inhalte</b> (allgemein für das Modul)	<ul style="list-style-type: none"> <li>• Einleitung und Motivation</li> <li>• Grundlagen Kommunikation</li> <li>• Interprozesskommunikation</li> <li>• Prozesse</li> <li>• Synchronisationsalgorithmen</li> <li>• Koordinationsalgorithmen</li> <li>• Kurze Einführung in Architekturmodelle und Software Architekturen</li> </ul>		
<b>8. Unterlagen (Skripte, Literatur)</b>	<ul style="list-style-type: none"> <li>• Folien</li> <li>• Tanenbaum, M.v.Steen: Verteilte Systeme: Grundlagen und Paradigmen, Pearson Studium</li> <li>• G. Coulouris, J. Dollimore, T. Kindberg: Verteilte Systeme: Konzepte und Design, Pearson Studium</li> <li>• S. Dustdar, H. Gall, M. Hauswirth: Software-Architekturen für Verteilte Systeme, Springer Verlag</li> </ul>		
<b>9. Lehrform/en</b> (ggf. Anzahl der zu besuchenden LV mit Lehrform)	Vorlesung mit integrierter Übung, Vorträge der Studierenden		
<b>10. Semesterempfehlung</b>	2		
<b>11. Häufigkeit des Angebots</b>	Jährlich		
<b>12. Inhaltliche Voraussetzungen</b>	Bachelor Informatik		
<b>13. Arbeitsaufwand (gesamt)</b>	180 Stunden (Präsenz: 56h, Selbststudium: 124h)		
<b>14. Anzahl der LP</b>	6		
<b>15. Voraussetzungen für die Vergabe von LP/ECTS</b>	mdl. Prüfung		
<b>16. Prüfung</b>	Modulgesamtprüfung Verteilte Systeme		
<b>17. Anmeldeformalitäten</b>	Anmeldung zur Prüfung in STUDIS		
	Modulgesamtprüfung Verteilte Systeme	SWS	4
<b>Summe:</b>			LP: 6





<b>1. Modultitel</b>	Modellbasierte Sicherheitsanalyse
<b>2. Englischer Modultitel</b>	Safety Analysis
<b>3. Modulbereich</b>	<input type="checkbox"/> Pflichtbereich <input checked="" type="checkbox"/> Wahlbereich
<b>4. Fachgebiet</b>	Formale Methoden und IT-Sicherheit
<b>5. Modulbeauftragte/r</b>	Prof. Dr. Wolfgang Reif
<b>6. Lernziele/Lernergebnis</b> (allgemein für das Modul)	<p>Studenten erlangen in diesem Modul:</p> <ul style="list-style-type: none"> <li>• Vertiefte Kenntnisse über Methoden und Techniken der formalen Analyse sicherheitskritischer Systeme</li> <li>• Erfahrung in der Modellierung mit Temporallogik</li> <li>• Erfahrung im Einsatz von Modelchecking</li> <li>• Kenntnisse in der quantitativen Analyse von Systemen mit antagonistischen Sicherheitszielen</li> </ul>
<b>7. Inhalte</b> (allgemein für das Modul)	<p>Safety-critical systems are expected to operate safely under regular circumstances as well as in many degraded situations. In the latter case, these systems have to cope with one or more components that are not working as specified, while at the same time they have to guarantee that no harm is done to people or the environment. A wide variety of traditional safety analysis techniques (such as Fault Tree Analysis or Failure Modes and Effects Analysis) help safety engineers in systematically analyzing a system: They dissect the system to determine possible (combinations of) component failures that might result in an occurrence of a dangerous situation, called a hazard. However, the functionality provided by safety-critical systems is becoming increasingly complex, therefore requiring the development of more sophisticated analysis techniques to analyze the system behavior under both regular and degraded situations. Additionally, software is becoming an increasingly important factor for the innovation of safety-critical systems; more and more safety-critical hardware is replaced by software. However, software development is complex and error-prone and is thus likely to introduce systematic errors that have the potential of violating safety requirements.</p> <p>This course gives an introduction to traditional approaches in safety analysis as well as leading edge analysis techniques that are based on formal methods. Additionally, the course shows how the occurrence probabilities of the hazards can be computed and how antagonistic safety goals can be balanced to find an optimal trade-off.</p>
<b>8. Unterlagen (Skripte, Literatur)</b>	<ul style="list-style-type: none"> <li>• N. Leveson. Safeware–System Safety and Computers. Addison-Wesley 1995</li> <li>• N. Storey. Safety-Critical Computer Systems. Addison-Wesley 1996</li> <li>• E. Clarke, O. Grumberg, D. Peled. Model Checking. MIT Press 2001</li> <li>• D. W. Vesley, D. J. Dugan, J. Fragole, J. MinarickIII, J. Railsback. Fault Tree Handbook with Aerospace Applications. NASA Office of Safety 2002</li> </ul>
<b>9. Lehrform/en</b> (ggf. Anzahl der zu besuchenden LV mit Lehrform)	Vorlesung
<b>10. Semesterempfehlung</b>	2

<b>11. Häufigkeit des Angebots</b>	Jährlich		
<b>12. Inhaltliche Voraussetzungen</b>			
<b>13. Arbeitsaufwand (gesamt)</b>	150		
<b>14. Anzahl der LP</b>	5		
<b>15. Voraussetzungen für die Vergabe von LP/ECTS</b>	Klausur und mündliche Prüfung		
<b>16. Prüfung</b>	Modulgesamtprüfung Modellbasierte Sicherheitsanalyse		
<b>17. Anmeldeformalitäten</b>	Anmeldung zur Prüfung in STUDIS		
	Modulgesamtprüfung Modellbasierte Sicherheitsanalyse	SWS	3
<b>Summe:</b>			LP: 5

<b>1. Modultitel</b>	Modellbasierte Softwareentwicklung
<b>2. Englischer Modultitel</b>	Model-based Software Engineering
<b>3. Modulbereich</b>	<input type="checkbox"/> Pflichtbereich <input checked="" type="checkbox"/> Wahlbereich
<b>4. Fachgebiet</b>	Softwaretechnik
<b>5. Modulbeauftragte/r</b>	Prof. Dr. Alexander Knapp
<b>6. Lernziele/Lernergebnis</b> (allgemein für das Modul)	<p>Studierende erlangen in diesem Modul:</p> <ul style="list-style-type: none"> <li>• Vertiefte Kenntnisse in der Modellierung von Software, in Modellierungstechniken und Modelltransformationen</li> <li>• Fähigkeiten im Umgang mit technischen Umgebungen zur modellbasierten Entwicklung und zu ihrer technischen Bewertung</li> <li>• Verständnis für das Verhältnis von Modell und Implementierung</li> </ul>
<b>7. Inhalte</b> (allgemein für das Modul)	<p>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 an Hand der Themenschwerpunkte: Modellierung, Metamodellierung, Transformationen und Codegenerierung.</p> <p>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)</p>
<b>8. Unterlagen (Skripte, Literatur)</b>	<ul style="list-style-type: none"> <li>• David S. Frankel. Model Driven Architecture. OMG Press, 2003.</li> <li>• Jack Greenfield, Keith Short. Software Factories. Wiley, 2004.</li> <li>• Volker Gruhn, Daniel Pieper, Carsten Röttgers. MDA. Springer, 2006.</li> <li>• Georg Pietrek, Jens Trompeter (Hgg.). Modellgetriebene Softwareentwicklung. entwickler.press, 2007.</li> <li>• Chris Raistrick, Paul Francis, John Wright, Colin Carter, Ian Wilkie. Model Driven Architecture with Executable UML. Cambridge University Press, 2004.</li> <li>• Thomas Stahl, Markus Völter, Sven Efftinge, Arno Haase. Modellgetriebene Softwareentwicklung. dpunkt.verlag, 2007.</li> </ul>
<b>9. Lehrform/en</b> (ggf. Anzahl der zu besuchenden LV mit Lehrform)	Vorlesung mit integrierter Übung
<b>10. Semesterempfehlung</b>	2

<b>11. Häufigkeit des Angebots</b>	Jährlich		
<b>12. Inhaltliche Voraussetzungen</b>	-		
<b>13. Arbeitsaufwand (gesamt)</b>	150 (Präsenz: 42h, Selbststudium: 108h)		
<b>14. Anzahl der LP</b>	5		
<b>15. Voraussetzungen für die Vergabe von LP/ECTS</b>	Mündl. Prüfung		
<b>16. Prüfung</b>	Modulgesamtprüfung Modellbasierte Softwareentwicklung		
<b>17. Anmeldeformalitäten</b>	Anmeldung zur Prüfung in STUDIS		
	Modulgesamtprüfung Modellbasierte Softwareentwicklung	SWS	3
<b>Summe:</b>			LP: 5

<b>1. Modultitel</b>	Modellierung, Spezifikation und Verifikation reaktiver Systeme
<b>2. Englischer Modultitel</b>	Modelling, Specification and Verification of Reactive Systems
<b>3. Modulbereich</b>	<input type="checkbox"/> Pflichtbereich <input checked="" type="checkbox"/> Wahlbereich
<b>4. Fachgebiet</b>	Formale Methoden und IT-Sicherheit
<b>5. Modulbeauftragte/r</b>	Prof. Dr. Alexander Knapp
<b>6. Lernziele/Lernergebnis</b> (allgemein für das Modul)	<p>Studierende erlangen in diesem Modul:</p> <ul style="list-style-type: none"> <li>• Vertiefte Kenntnisse in der Modellierung und Spezifikation reaktiver Systeme sowie deren Verifikation</li> <li>• Fähigkeiten im Umgang mit verschiedenen Modellierungsansätzen und -werkzeugen</li> <li>• Verständnis für die spezifischen Probleme reaktiver und paralleler Systeme</li> </ul>
<b>7. Inhalte</b> (allgemein für das Modul)	<p>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.</p> <p>Schlagwörter: Transitionssysteme, Sicherheits- und Lebendigkeitseigenschaften; strombasierte Spezifikationen; nebenläufige Termersetzungssysteme (z. B. Maude); Prozessalgebren (z. B. Calculus of Communicating Systems, Communicating Sequential Processes, <math>\pi</math>-Kalkül), Simulation und Bisimulation; Temporallogik, symbolisches Model checking, automatenbasiertes Model checking, Abstraktion; Temporal Logic of Actions, Verfeinerung</p>

<b>8. Unterlagen (Skripte, Literatur)</b>	<ul style="list-style-type: none"> <li>• Luca Aceto, Anna Ingólfssdóttir, Kim G. Larsen, Jiří Srba. Reactive Systems: Modelling, Specification and Verification. Cambridge University Press, 2007.</li> <li>• Christel Baier, Joost-Pieter Katoen. Principles of Model Checking. MIT Press, 2008.</li> <li>• Klaus Schneider. Verification of Reactive Systems. Springer, 2003.</li> <li>• Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, Carolyn Talcott. All About Maude - A High-Performance Logical Framework. How to Specify, Program, and Verify Systems in Rewriting Logic. Lect. Notes Comp. Sci. 4350, Springer, 2007.</li> <li>• Robin Milner. Communication and Concurrency. Prentice Hall, 1989.</li> <li>• Robin Milner. Communicating and Mobile Systems: The <math>\pi</math>-Calculus. Cambridge University Press, 1999.</li> <li>• Fred Kröger, Stephan Merz. Temporal Logic and State Systems. Springer, 2008.</li> <li>• Edmund M. Clarke, Jr., Orna Grumberg, Doron A. Peled. Model Checking. MIT Press, 1999.</li> <li>• Leslie Lamport. Specifying Systems. Addison-Wesley, 2003.</li> </ul>		
<b>9. Lehrform/en</b> (ggf. Anzahl der zu besuchenden LV mit Lehrform)	Vorlesung mit integrierter Übung		
<b>10. Semesterempfehlung</b>	2		
<b>11. Häufigkeit des Angebots</b>	Jährlich		
<b>12. Inhaltliche Voraussetzungen</b>	Formale Methoden		
<b>13. Arbeitsaufwand (gesamt)</b>	180 (Präsenz: 56h, Selbststudium: 124h)		
<b>14. Anzahl der LP</b>	6		
<b>15. Voraussetzungen für die Vergabe von LP/ECTS</b>	Mündl. Prüfung		
<b>16. Prüfung</b>	Modulgesamtprüfung Modellierung, Spezifikation und Verifikation reaktiver Systeme		
<b>17. Anmeldeformalitäten</b>	Anmeldung zur Prüfung in STUDIS		
	Modulgesamtprüfung Modellierung, Spezifikation und Verifikation reaktiver Systeme	SWS	4
<b>Summe:</b>			LP: 6

<b>1. Modultitel</b>	Testen
<b>2. Englischer Modultitel</b>	Testing
<b>3. Modulbereich</b>	<input type="checkbox"/> Pflichtbereich <input checked="" type="checkbox"/> Wahlbereich
<b>4. Fachgebiet</b>	Softwaretechnik
<b>5. Modulbeauftragte/r</b>	Prof. Dr. Alexander Knapp
<b>6. Lernziele/Lernergebnis</b> (allgemein für das Modul)	<p>Studierende erlangen in diesem Modul:</p> <ul style="list-style-type: none"> <li>• Vertiefte Kenntnisse im Testen von Software, der zugrundeliegenden Techniken und dem Einsatz im Softwareentwicklungsprozess</li> <li>• Fähigkeiten zum Einsatz von Testtechniken und -werkzeugen für verschiedene Anwendungstypen und deren Bewertung</li> <li>• Verständnis für die Möglichkeiten und Grenzen des Softwaretests</li> </ul>
<b>7. Inhalte</b> (allgemein für das Modul)	<p>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.</p> <p>Schlagwörter: Software-Qualität, Testprozess, funktionales Testen, strukturelles Testen, diversifizierendes Testen, modellbasiertes Testen, Testen eingebetteter Software (Konformanztesten), Testen objektorientierter Software, Integrationstest, Testausführung</p>
<b>8. Unterlagen (Skripte, Literatur)</b>	<ul style="list-style-type: none"> <li>• Robert V. Binder. Testing Object-Oriented Systems: Models, Patterns, and Tools. Addison-Wesley, 1999.</li> <li>• Daniel Galin. Software Quality Assurance: From Theory to Implementation. Addison Wesley, 2003.</li> <li>• Peter Liggesmeyer. Software-Qualität: Testen, Analysieren und Verifizieren von Software. Spektrum Akademischer Verlag, 2002.</li> <li>• Mauro Pezzè, Michal Young. Software Testing and Analysis: Process, Principles and Techniques. Wiley &amp; Sons, 2007.</li> <li>• Andreas Spillner, Tilo Linz. Basiswissen Softwaretest. dpunkt.verlag, 3. Aufl., 2005.</li> <li>• Mark Utting, Bruno Legeard. Practical Model-Based Testing. Morgan Kaufmann, 2007.</li> </ul>
<b>9. Lehrform/en</b> (ggf. Anzahl der zu besuchenden LV mit Lehrform)	Vorlesung mit integrierten Übungen
<b>10. Semesterempfehlung</b>	2
<b>11. Häufigkeit des Angebots</b>	Jährlich
<b>12. Inhaltliche Voraussetzungen</b>	-
<b>13. Arbeitsaufwand (gesamt)</b>	150

<b>14. Anzahl der LP</b>	5		
<b>15. Voraussetzungen für die Vergabe von LP/ECTS</b>	Mündl. Prüfung		
<b>16. Prüfung</b>	Modulgesamtprüfung Testen		
<b>17. Anmeldeformalitäten</b>	Anmeldung zur Prüfung in STUDIS		
	Modulgesamtprüfung Testen	SWS	3
<b>Summe:</b>			LP: 5



1. <b>Modultitel</b>	Datenbanksysteme und moderne CPU-Architekturen		
2. <b>Englischer Modultitel</b>	Database Systems on Modern CPU Architectures		
3. <b>Modulbereich</b>	<input type="checkbox"/> Pflichtbereich	<input checked="" type="checkbox"/> Wahlbereich	
4. <b>Fachgebiet</b>	Datenbanksysteme		
5. <b>Modulbeauftragte/r</b>	Prof. Dr. Thomas Neumann		
6. <b>Lernziele/Lernergebnis</b> (allgemein für das Modul)	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.		
7. <b>Inhalte</b> (allgemein für das Modul)	Implementation of Database Systems with respect to modern hardware: Storage Layer, Access Paths, Transactions, Set-Oriented Query Processing, Algebraic Operators		
8. <b>Unterlagen (Skripte, Literatur)</b>	Vorlesungsfolien, sowie Theo Härder, Erhard Rahm. <i>Datenbanksysteme: Konzepte und Techniken der Implementierung</i> . Springer, Berlin; 2nd ed.		
9. <b>Lehrform/en</b> (ggf. Anzahl der zu besuchenden LV mit Lehrform)	Vorlesung (2) + Übung (2)		
10. <b>Semesterempfehlung</b>	ab 2. Semester (Master)		
11. <b>Häufigkeit des Angebots</b>	jährlich		
12. <b>Inhaltliche Voraussetzungen</b>	Datenbanksysteme		
13. <b>Arbeitsaufwand (gesamt)</b>	180 (Präsenz: 56h, Selbststudium: 124h)		
14. <b>Anzahl der LP</b>	6		
15. <b>Voraussetzungen für die Vergabe von LP/ECTS</b>	Mündl. Prüfung		
16. <b>Prüfung</b>	Modulgesamtprüfung Datenbanksysteme und moderne CPU-Architekturen		
17. <b>Anmeldeformalitäten</b>	Anmeldung zur Prüfung in STUDIS		
	Modulgesamtprüfung Datenbanksysteme und moderne CPU-Architekturen	SWS	4
<b>Summe:</b>			LP: 6



<b>1. Modultitel</b>	Seminar zu Compilerbau		
<b>2. Englischer Modultitel</b>	Compiler Construction Seminar		
<b>3. Modulbereich</b>	<input type="checkbox"/> Pflichtbereich	<input checked="" type="checkbox"/> Wahlbereich	
<b>4. Fachgebiet</b>	Softwaretechnik		
<b>5. Modulbeauftragte/r</b>	Prof. Dr. B. Bauer		
<b>6. Lernziele/Lernergebnis</b> (allgemein für das Modul)	<p>Studierende erlangen in diesem Modul:</p> <ul style="list-style-type: none"> <li>• Vertiefte Kenntnisse in der Entwicklung von Compilern</li> <li>• Verständnis der lexikalischen, syntaktischen, semantischen Analyse von Programmen, Zwischensprachen, Codeerzeugung, sowie Optimierungen</li> <li>• Erfahrung in der Verwendung von Compilerbauwerkzeugen</li> <li>• Selbständiges wissenschaftliches Arbeiten und Präsentieren</li> </ul>		
<b>7. Inhalte</b> (allgemein für das Modul)	<p>Die Themen des Seminars gliedern sich in folgende Überkapitel:</p> <ul style="list-style-type: none"> <li>• Einleitung</li> <li>• Lexikalische Analyse</li> <li>• Syntaktische Analyse</li> <li>• Semantische Analyse</li> <li>• Zwischencodeerzeugung</li> <li>• Laufzeitumgebung</li> <li>• Übersetzung objektorientierter Programmiersprachen</li> </ul>		
<b>8. Unterlagen (Skripte, Literatur)</b>	<ul style="list-style-type: none"> <li>• <b>Compiler: Prinzipien, Techniken und Werkzeuge</b>, Ravi Sethi, Jeffrey D. Ullman, Alfred V. Aho, Monica S. Lam., Pearson, 2008</li> </ul>		
<b>9. Lehrform/en</b> (ggf. Anzahl der zu besuchenden LV mit Lehrform)	Seminar		
<b>10. Semesterempfehlung</b>	1		
<b>11. Häufigkeit des Angebots</b>	Jährlich		
<b>12. Inhaltliche Voraussetzungen</b>	--		
<b>13. Arbeitsaufwand (gesamt)</b>	90 Stunden		
<b>14. Anzahl der LP</b>	3		
<b>15. Voraussetzungen für die Vergabe von LP/ECTS</b>	Vortrag und Ausarbeitung		
<b>16. Prüfung</b>	Modulgesamtprüfung:Seminar Compilerbau		
<b>17. Anmeldeformalitäten</b>	Anmeldung zur Prüfung in STUDIS		
	Modulgesamtprüfung Seminar Compilerbau	SWS	2
<b>Summe:</b>			LP: 3



<b>1. Modultitel</b>	Seminar zu Avionic Software Engineering
<b>2. Englischer Modultitel</b>	Avionic Software Engineering Seminar
<b>3. Modulbereich</b>	<input type="checkbox"/> Pflichtbereich <input checked="" type="checkbox"/> Wahlbereich
<b>4. Fachgebiet</b>	Softwaretechnik
<b>5. Modulbeauftragte/r</b>	Prof. Dr. B. Bauer
<b>6. Lernziele/Lernergebnis</b> (allgemein für das Modul)	Studierende erlangen in diesem Modul: <ul style="list-style-type: none"> <li>• Vertiefte Kenntnisse in der Entwicklung von Software Engineering Konzepte für Avionic Systeme</li> <li>• Verständnis der relevanten Technologien und Standards</li> <li>• Selbständiges wissenschaftliches Arbeiten und Präsentieren</li> </ul>
<b>7. Inhalte</b> (allgemein für das Modul)	Die Themen des Seminars gliedern sich in folgende Überkapitel: <ul style="list-style-type: none"> <li>• Software Qualifikation von fliegenden Systemen nach Do-178B</li> <li>• Software Entwicklung nach dem V-Modell XT</li> <li>• Software Entwicklung nach dem Standard ARINC653</li> <li>• Traceability von den Anforderungen in die Software bis zu den Tests</li> <li>• Software Entwicklung mit der Programmiersprache Ada (83, 95, 2005)</li> <li>• Einsatz der unterschiedlichen Bus- und Protokoll-Systeme in einem Avionik System</li> <li>• Human Machine Interface (HMI) Definition nach ARINC739</li> <li>• Teststufen innerhalb der Entwicklung von Avionik Systemen</li> <li>• Anforderungsanalyse und -Management innerhalb der Entwicklung von Avionik Systemen</li> </ul>
<b>8. Unterlagen (Skripte, Literatur)</b>	<b>Einstiegsliteratur:</b> <ul style="list-style-type: none"> <li>• V-Modell XT Standard</li> <li>• DO-178 B Standard</li> <li>• Ada 83, 95, 2005 Sprachspezifikation</li> <li>• ARINC 749</li> </ul>
<b>9. Lehrform/en</b> (ggf. Anzahl der zu besuchenden LV mit Lehrform)	Seminar
<b>10. Semesterempfehlung</b>	1
<b>11. Häufigkeit des Angebots</b>	Jährlich
<b>12. Inhaltliche Voraussetzungen</b>	--
<b>13. Arbeitsaufwand (gesamt)</b>	90 Stunden
<b>14. Anzahl der LP</b>	3
<b>15. Voraussetzungen für die Vergabe von LP/ECTS</b>	Vortrag und Ausarbeitung
<b>16. Prüfung</b>	Modulgesamtprüfung:Seminar Avionic Software Engineering
<b>17. Anmeldeformalitäten</b>	Anmeldung zur Prüfung in STUDIS
Modulgesamtprüfung Seminar Avionic Software Engineering	
SWS	
2	

<b>Summe:</b>		LP: 3
---------------	--	-------

<b>1. Modultitel</b>	Führungskompetenz
<b>2. Englischer Modultitel</b>	
<b>3. Modulbereich</b>	<input type="checkbox"/> Pflichtbereich <input checked="" type="checkbox"/> Wahlbereich
<b>4. Fachgebiet</b>	Softskills
<b>5. Modulbeauftragte/r</b>	
<b>6. Lernziele/Lernergebnis</b> (allgemein für das Modul)	<p>Studierende erhalten in diesem Modul:</p> <ul style="list-style-type: none"> <li>● Einen Einblick in die Grundbausteine der Führung</li> <li>● Die Möglichkeit, den eigenen Führungsstil zu erkennen und zu entwickeln</li> <li>● Erfahrung in der Motivation anderer</li> <li>● Feedback zum Verhältnis von Selbstbild und Fremdbild</li> </ul>
<b>7. Inhalte</b> (allgemein für das Modul)	<p>In diesem Modul werden die Grundbausteine kompetenter Führung erarbeitet. Hierbei lernen die Studierenden ihren persönlichen Führungsstil kennen und entwickeln Vertrauen in ihre individuelle Art, andere Menschen zu motivieren, Aufgaben zu delegieren und ihre führungspezifische Kommunikation auszubauen. Sie kommen in Kontakt mit ihrem individuellen Führungspotential und erfahren ihre Wirkung auf andere.</p> <p><b>Leitfragen</b></p> <ul style="list-style-type: none"> <li>● Wie entwickle ich eine Vertrauensbasis zu meinen Kollegen und Mitarbeitern?</li> <li>● Wie erkenne ich Konflikte in meinem Team und wie gehe ich damit um?</li> <li>● Was für ein Führungstyp bin ich?</li> <li>● Wie kommuniziere ich motivierend?</li> <li>● Wie gewinne ich andere für meine Ziele?</li> <li>● Wie gehe ich mit Macht um?</li> <li>● In welchen Punkten differiert mein Selbstbild vom Fremdbild?</li> </ul>
<b>8. Unterlagen (Skripte, Literatur)</b>	
<b>9. Lehrform/en</b> (ggf. Anzahl der zu besuchenden LV mit Lehrform)	Praxisorientiertes Seminarkonzept, Impulsvorträge, Gruppenarbeit, begleitete Rollen- und Interaktionsspiele, Reflexion der gewonnenen Erfahrungen
<b>10. Semesterempfehlung</b>	3
<b>11. Häufigkeit des Angebots</b>	Jährlich
<b>12. Inhaltliche Voraussetzungen</b>	
<b>13. Arbeitsaufwand (gesamt)</b>	90 (Präsenz: 30h, Selbststudium: 60h)
<b>14. Anzahl der LP</b>	3
<b>15. Voraussetzungen für die Vergabe von LP/ECTS</b>	Leistungsnachweis
<b>16. Prüfung</b>	Modulgesamtprüfung Führungskompetenz
<b>17. Anmeldeformalitäten</b>	Anmeldung zur Prüfung in STUDIS

	Modulgesamtprüfung Führungskompetenz	SWS	2
<b>Summe:</b>			LP: 3