Software Technology

– Module Overview

Autor:	Marcus Deininger
Version:	1.0
Date:	19. December 2019

1 Software Technoloy – Module Overview

1. Year – 1. Semester (March – August)

Compulsary Modules			
Short	**Module**	**ECTS**	**Type**
CPL	Concepts of Programming Languages	5	Lecture
DAB	Advanced Topics in Databases	6	Lecture
SPM	Advanced Topics in Software Project Management	6	Lecture
SWE	Advanced Topics in Software Engineering	6	Lecture
ICT	Intercultural Training (Part 1)	1	Project
EL1	Elective Module I	6	Lecture

Current Elective Modules are “Advanced Topics in Algorithms and Data Structures“ and „Computer Vision“

1. Year – 2. Semester (October – February)

Compulsary Modules			
Short	**Module**	**ECTS**	**Type**
MWT	Middleware Technology	6	Lecture
SOP	Software Project	8	Project
SVV	Software Verification and Validation	3	Lecture
SYD	System Design	6	Lecture
ICT	Intercultural Training (Part 2)	1	Project
EL1	Elective Module I	6	Lecture

Current Elective Modules are “Business Intelligence“ and „Business Process Technologies“

2. Year – 1. Semester

Compulsary Modules			
Short	**Module**	**ECTS**	**Type**
MT	Master Thesis	30	Thesis

1.1 Concepts of Programming Languages

Course	Master Software Technology
Name of Module	Concepts of Programming Languages
Abbreviation	CPL
Semester	1
Responsible	Prof. Dr. Heusch
Lecturers	Prof. Dr. Heusch, Prof. Dr. Padó
Method of Teaching	Lectures with Exercises, Reading Assignments, Assignments
European Credit Transfer Points	5 ECTS Points
Weekly Contact Hours	4h
Student Work Load	150h Total: • Lectures (68h) • Self Studies (82h)
Necessary Previous Knowledge	Comprehensive knowledge of Java
Final Knowledge and Skills	*Knowledge and understanding* On completion the student knows the different language paradigms and concepts, especially in • Procedural Programming • Functional Programming • Object Oriented Programming • Logic Programming *Disciplinary / professional skills* On completion the student is able to select a fitting paradigm and programming language for a given problem and to rate the implications of this decision.
Index	• The early days: FORTRAN, COBOL, PL/1, Assembler • Recursion, Functions and Lambda Calculus: Lisp, Scheme, Forth • The Algol-Languages: Algol, Pascal, PL/SQL, C • Object-oriented Programming:, C++, Java, Scala • Weakly typed languages: Perl, Python, JavaScript, Ruby • Programming in Logic: PROLOG • Special Purpose Programming Languages: TeX, M4
Method and Extent of Examination	Written examination, 120 minutes
Pre-Exam Require-	Assignments

ments	
Recommended Literature (Excerpt)	• Abelson, H., J. Sussmann: Structure and Interpretations of Programming Languages. MIT-Press/ McGraw-Hill, 1996. • Clocksin, W., C. Mellish: Programming in Prolog, Springer, 2003. • Mitchell, J.: Concepts in Programming Languages. Cambridge University Press, 2001. • Sebesta, R.: Concepts of Programming Languages. Addison-Wesley 2003. • Various international research papers (distributed in class)

1.2 Databases II

Course	Master Software Technology
Name of Module	Databases II (Advanced Topics in Databases)
Abbreviation	DAB
Semester	1
Responsible	Prof. Koch
Lecturers	Prof. Koch, Prof. Dr. Kramer
Method of Teaching	Lecture with theoretical and practical exercises; independent group project with presentation
European Credit Transfer Points	6 ECTS Points
Weekly Contact Hours	4h (3h Lectures + 1h Exercises/Presentations)
Student Work Load	180h Total: • Lectures (68h) • Self Studies (112h)
Necessary Previous Knowledge	Data structures/algorithms; Bachelor level understanding of file systems, computer architecture, and databases; Entity Relationship Modeling; basic knowledge of the relational model and SQL
Final Knowledge and Skills	Knowledge and understanding On completion the student has a deeper understanding of DBMS functionality and in particular of modern system approaches. He or she has practical experience with at least one relational database system and insight into current database research issues. Disciplinary / professional skills On completion the student is able to evaluate strengths and weaknesses of database and transaction processing systems and to make informed decisions about different situations of database usage in

	practical projects within enterprise contexts.
Index	• Review of principles of relational databases, advanced features of SQL, the MySQL DBMS • Database programming (ODBC, SQL/CLI, JDBC, Embedded SQL, Dynamic SQL, SQLJ) • Transaction management: review of basic properties, distributed and nested transactions, sagas, 2 phase and 3 phase commit protocol, long transactions, architecture and functionality of transaction processing systems • Recovery: logging, checkpointing, savepointing, recovery after software and hardware failures, backup methods • Concurrency control: 2 phase locking, isolation levels, timestamp and optimistic protocols • Distributed databases: data fragmentation, replication, and allocation techniques; distributed recovery and concurrency control • Mobile databases: architecture, data replication, transaction processing, performance • Object-oriented and object-relational databases, comparison to relational systems
Method and Extent of Examination	Written examination, 120 minutes
Pre-Exam Requirements	Successful group project
Recommended Literature (Excerpt)	• Bernstein, P., E. Newcomer: Principles of Transaction Processing for the System Professional. Morgan Kaufmann, 1997. • Cattell, R.G.G.: Object Data Management, Addison-Wesley, 1994. • Ceri, S., G. Pelagatti: Distributed Databases, Principles and Systems. McGraw-Hill, 1984. • Connolly, T.M., C.E. Begg, A.D. Strachan: Database Systems, A Practical Approach to Design, Implementation and Management. Addison-Wesley, 2001. • Date, C.J.: An Introduction to Database Systems. Addison Wesley, 1999. • Elmasri, R., S. Navathe: Fundamentals of Database Systems. Addison Wesley 2004. • Gray, J., A. Reuter: Transaction Processing, Concepts and Techniques. Morgan Kaufmann, 1993. • Ozsu, M.T., P. Valduriez: Principles of Distributed Database Systems. Prentice Hall, 1999. • Stonebraker, S., D. Moore, P. Brown: Object-Relational DBMSs. Morgan Kaufmann, 1998. • Various international research papers (distributed in class)

1.3 Software Project Management II

Course	Master Software Technology
Name of Module	Software Project Management II (Advanced Topics in Software Project Management)
Abbreviation	SPM
Semester	1
Responsible	Prof. Dr. Kramer
Lecturers	Prof. Dr. Deininger, Prof. Dr. Höß, Prof. Dr. Kramer, Prof. Dr. Lückemeyer
Method of Teaching	Lecture with exercises, students' presentations, applying project management software
European Credit Transfer Points	6 ECTS Points
Weekly Contact Hours	4h (3h Lectures + 1h Exercises/Presentations)
Student Work Load	180h Total: • Lectures (68h) • Self Studies (112h)
Necessary Previous Knowledge	• Software Project Management (Bachelor Level) • Experience in (small) software projects, either at the university or in industry
Final Knowledge and Skills	Knowledge and understanding On completion the student knows different methods for estimating efforts and costs of software projects. He or she understands the underlying principles of project management software. He or she is well aware of agile approaches, quality assurance, risk management and maturity models, their usage and their benefits. Disciplinary / professional skills On completion the student is able to choose an adequate overall approach for projects of different kinds and sizes and to plan and to control projects using project plans if required. He or she is able to select and to use appropriate cost estimation methods and project management software in practical projects. He or she is able to apply methods for quality control and for risk management and to use maturity models for improving processes.
Index	• Brief recap of software project management basics (e.g. work break down structure, project organization) • Methods for planning and controlling projects • Estimation methods: efforts, costs • Network planning techniques

	• Project management software • Quality assurance • Risk management • Maturity models
Method and Extent of Examination	Oral Examination, 20 minutes
Pre-Exam Requirements	Individual presentation in class, team submissions to selected exercises
Recommended Literature (Excerpt)	• Futrell, R.T., D.F. Shafer, L.I. Shafer: Quality Software Project Management. Software Quality Institute Series. Prentice Hall, 2002. • P. Bourque and R.E. Fairley, eds.: Guide to the Software Engineering Body of Knowledge, Version 3.0 (SWEBOK); IEEE Computer Society, 2014, www.swebok.org. • PMI Standards Committee: A Guide to the Project Management Body of Knowledge (PMBOK); Project Management Institute, 5th edition, 2013. • subject specific additional literature, project management software

1.4 Software Engineering II

Course	Master Software Technology
Name of Module	Software Engineering II (Advanced Topics in Software Engineering)
Abbreviation	SWE
Semester	1
Responsible	Prof Dr. Wanner
Lecturers	Prof. Dr. Deininger, Prof. Dr. Wanner
Method of Teaching	Lecture with Exercises, Reading Assignments, Assignments
European Credit Transfer Points	6 ECTS Points
Weekly Contact Hours	4h (3h Lectures + 1h Exercises/Presentations)
Student Work Load	180h Total: • Lectures (68h) • Self Studies (112h)
Necessary Previous Knowledge	• Programming experience • Software Engineering (Bachelor Level)

Final Knowledge and Skills	Knowledge and understanding On completion the student has a deeper understanding of the software development process. He or she knows about current approaches in application performance engineering, testing, quality-, software-architecture-, change and configuration management. Disciplinary / professional skills On completion the student is able for a given project type to identify the most suitable approach to software development or procurement. He or she is able to use generic and generative approaches to generate applications out of an extended analysis model. He or she can apply analysis pattern for modeling complex software systems and is able to evaluate, select and use modern development techniques and application performance engineering in practical projects within enterprise contexts. He or she can use software-architecture management to check planned and actual architecture and is able to introduce quality- and change management in a project.
Index	• Design Techniques: Architectural Principles and Patterns • Development Techniques: Model-Driven-Development, Component-Based-Development, Aspect-Oriented-Development, Cloud-Computing … • Test Techniques: Stubbing, Mocking, Test Code Injection, … • Build Management, Software-Architecture Management • Quality Management, Quality Assurance, Application Performance Engineering • Open Source Software Development
Method and Extent of Examination	Written examination, 120 minutes
Pre-Exam Requirements	Assignments
Recommended Literature (Excerpt)	• Brambilla, M., Cabot, J., Wimmer, M.: Model-Driven Software Engineering in Practice, Morgan & Claypool Publishers, 2012. • Fowler, M.: Refactoring, Addison Wesley. Boston, 2001. • Gamma, E., R. Helm, R. Johnson, J. Vlissides: Design Patterns: Elements of Reusable Object Oriented Software. Addison-Wesley Longman, 1997. • Krzysztof Czarnecki, Ulrich W. Eisenecker: Generative Programming, Addison-Wesley, 2000 • Palmer, S.R., J.M. Felsing: A Practical Guide to Feature-Driven Development. Prentice Hall, 2002. • Utting, M., B. Legeard: Practical Model-Based Testing. Elsevier, Morgan Kaufmann, 2007

1.5 Intercultural Training

Course	Master Software Technology
Name of Module	Intercultural Training
Abbreviation	ICT
Semester	1 and 2

Note: Intercultural Training 1 and 2 is held as a block event at the beginning of the first and second semester and is mainly used to integrate new students and especially lateral entrants into the existing group. For this reason the contents of the two modules overlap.

Course	Master Software Technology
Name of Module	Intercultural Training (Part 1)
Abbreviation	ICT1
Semester	1
Responsible	Course Director Software Technology
Lecturers	Dr. Melinda Madew
Method of Teaching	Simulated Activities, Case Studies, Critical Incidents, Film Analyses, Role Playing, Lecture Input, Group Discussions, Reports.
European Credit Transfer Points	1 ECTS Point
Weekly Contact Hours	1h (held as a Block-course)
Student Work Load	30h Total: • Lectures (30h) • Self Studies (none)
Necessary Previous Knowledge	none
Final Knowledge and Skills	Knowledge and understanding On completion the student knows the theoretical bases of intercultural discipline and the rationale behind intercultural learning. He or she has a deeper understanding of communications, decision making, and cultural differences. Also, he or she knows better about their classmates and their cultural background. Disciplinary / professional skills On completion the student is able to sense cultural differences and adapt his communication and decisions to multi-cultural workplace.
Index	• Theoretical Bases: What is Culture? Understanding Concepts of Interculturality, Multiculturality, Diversity, Cultural Programming,

	Cultural Perception. • Working with Hofstede´s Dimensions of Culture: Individualist vs. Collectivist Culture; Dealing with Power and Hierarchy; Monolinear and Polylinear Culture; Gender and Culture. • Skills and Processes: Perceiving; Communicating; Managing Cultural Conflict; Coping with Diversity
Method and Extent of Examination	Individual Participation, Group Input, Reporting, Project Submission
Pre-Exam Requirements	Project Submission
Recommended Literature (Excerpt)	none

Course	Master Software Technology
Name of Module	Intercultural Training Part 2
Abbreviation	ICT2
Semester	2
Responsible	Course Director Software Technology
Lecturers	Dr. Melinda Madew
Method of Teaching	Simulated Activities, Case Studies, Critical Incidents, Film Analyses, Role Playing, Lecture Input, Group Discussions, Reports.
European Credit Transfer Points	1 ECTS Point
Weekly Contact Hours	1h (held as a Block-course)
Student Work Load	30h Total: • Lectures (30h) • Self Studies (none)
Necessary Previous Knowledge	none
Final Knowledge and Skills	Knowledge and understanding On completion new students know the theoretical bases of intercultural discipline and the rationale behind intercultural learning. He or she has a deeper understanding of communications, decision making, and cultural differences. All new students know better about their classmates and their cultural background. Disciplinary / professional skills On completion the student is able to sense cultural differences and adapt his communication and decisions to multi-cultural workplace.

Index	• Integration of Newcomers • Wrap-up for Newcomers: Theoretical Bases, Working with Hofstede's Dimensions of Culture, Skills and Processes • Further Skills and Processes: Perceiving; Communicating; Managing Cultural Conflict; Coping with Diversity
Method and Extent of Examination	Individual Participation, Group Input, Reporting, Project Submission
Pre-Exam Requirements	Project Submission
Recommended Literature (Excerpt)	none

1.6 Elective Module 1

Course	Master Software Technology
Name of Module	Elective Module 1
Abbreviation	EL1
Semester	1

1.6.1 Computer Vision (Additional ElectiveModule 1)

Course	Master Software Technology
Name of Module	Computer Vision (Additional ElectiveModule 1)
Abbreviation	CV
Semester	1
Responsible	Dr. Markus Enzweiler
Lecturers	Dr. Markus Enzweiler
Method of Teaching	Lecture with theoretical and practical exercises; independent group project with presentation
European Credit Transfer Points	6 ECTS Points
Weekly Contact Hours	4h (3h Lectures + 1h Exercises/Presentations)
Student Work Load	180h Total: • Lectures (68h)

	• Self Studies (112h)
Necessary Previous Knowledge	Programming experience, basic knowledge of linear algebra, calculus, and probability theory.
Final Knowledge and Skills	Knowledge and understanding On completion the student understands the fundamental concepts of computer vision, its mathematical foundations as well as its modern applications. He or she possesses detailed knowledge on topics such as image processing, object detection, scene recognition, stereo vision and motion analysis. Disciplinary / professional skills On completion the student is able to produce a theoretical concept and implement a practical solution to a problem involving computer vision and machine learning.
Index	• Applications of Computer Vision • Cameras and Optics • Image Filtering • Feature Detection • Model Fitting • Machine Learning • Object Detection • Motion Analysis • Stereo Vision
Method and Extent of Examination	Written examination, 120 minutes
Pre-Exam Requirements	Assignments
Recommended Literature (Excerpt)	• Szeliski, R.: *Computer Vision: Algorithms and Applications*, Springer Science and Business Media, 2010 • Forsyth, D., J. Ponce, *Computer Vision: A Modern Approach 2nd edition,* Pearson, 2012 • Gonzalez, R., R. Woods, *Digital Image Processing*, Prentice Hall, 2008

1.6.2 Data Structures and Algorithms II

Course	Master Software Technology
Name of Module	Data Structures and Algorithms II (Advanced Topics in Data Structures and Algorithms)
Abbreviation	DSA

Semester	1
Responsible	Prof. Dr. Homberger
Lecturers	Prof. Dr. Heusch, Prof. Dr. Homberger
Method of Teaching	Lecture with theoretical and practical exercises; independent group project with presentation
European Credit Transfer Points	6 ECTS Points
Weekly Contact Hours	4h (3h Lectures + 1h Exercises/Presentations)
Student Work Load	180h Total: • Lectures (68h) • Self Studies (112h)
Necessary Previous Knowledge	Basic principles of data structures and algorithms
Final Knowledge and Skills	Knowledge and understanding On completion the student understands algorithms for complex optimization problems used in decision making and automated coordination of self-interested decision makers. Moreover the students know advanced data structures for the efficient implementation of these algorithms. He or she knows about application areas of these optimization methods and data structures like Electronic Business, and Advanced Planning Systems. *Disciplinary / professional skills* On completion the student is able to select and implement an appropriate algorithm for a given problem.
Index	• Metaheuristics • Parallelization of metaheuristics • Multi-criteria optimization • Decentralized optimization • Collaborative planning and coordination • Electronic negotiation
Method and Extent of Examination	Written examination, 120 minutes
Pre-Exam Requirements	Assignments
Recommended Literature (Excerpt)	• Alba, E.: Parallel metaheuristics. Wiley, Hoboken, New Jersey, 2005. • Dorigo M., T. Stützle: Ant colony optimization. MIT Press, Cambridge, Massachusetts, 2004. • Eiben, A.E., J.E. Smith: Introduction to evolutionary computing. Springer, Berlin, 2003.

	• Fink A., J. Homberger (2014): Decentralized multi-project scheduling. In: Schwindt C., J. Zimmermann (eds.): Handbook on Project Management and Scheduling Vol. 2, International Handbooks on Information Systems, Springer, 2014. • Jennings, N.R., P. Faratin, A.R. Lomuscio, S. Parsons, M. Woolridge, C. Sierra: Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation* 10, 199-215. 2001. • Klein, M., P. Faratin, H. Sayama, Y. Bar-Yam: Negotiating complex contracts. *Group Decision and Negotiation* 12, 111-125. 2003. • Stadtler, H.: A framework for collaborative planning and state-of-the-art. *OR spectrum* 31*, 5-30*. 2009. • Talbi, E.-G.: Metaheuristics – from design to implementation. John Wiley & Sons, Hoboken, New Jersey, 2009.

1.6.3 Additional Elective Module 1

Course	Master Software Technology
Name of Module	Additional Elective Module 1
Abbreviation	ADD1
Semester	1
Responsible	Course Director of Software Technology
Lecturers	depending on actual topic
Method of Teaching	depending on actual topic
European Credit Transfer Points	6 ECTS Points
Weekly Contact Hours	4h
Student Work Load	180h Total: • Lectures (68h) • Self Studies (112h)
Necessary Previous Knowledge	depending on actual topic
Final Knowledge and Skills	depending on actual topic
Index	depending on actual topic
Method and Extent of Examination	to be defined by the examination board

Pre-Exam Requirements	to be defined by the examination board
Recommended Literature (Excerpt)	depending on actual topic

1.7 Middleware Technology

Course	Master Software Technology
Name of Module	Middleware Technology
Abbreviation	MWT
Semester	2
Responsible	Prof. Dr. Wanner
Lecturers	Prof. Dr. Keller, Prof. Dr. Wanner
Method of Teaching	Lectures with Exercises, Lab Work (Weekly Assignments)
European Credit Transfer Points	6 ECTS Points
Weekly Contact Hours	4h (2h Lectures + 2h Exercises/Lab Sessions)
Student Work Load	180h Total: • Lectures (68h) • Self Studies (112h)
Necessary Previous Knowledge	• Operating Systems • Object Oriented Software Implementation • Networks
Final Knowledge and Skills	*Knowledge and understanding* On completion the student knows the different classes of middleware. He or she understands the functions and the services of an application server and knows the SOA approach and Enterprise Service Bus. *Disciplinary / professional skills* On completion the student is able to develop software components using the object oriented middleware (i.e. using CORBA, RMI, etc.) and to design, develop and deploy distributed Java EE applications using an application server and the services of this environment. He or she is able to design, implement and provide applications using the SOA and Enterprise Service Bus.
Index	• Types of middleware • Object oriented middleware (e.g. CORBA) • Message oriented middleware (e.g. JMS)

	• Types of Application Servers • Services of the application server (transaction service, security service, naming service, lifecycle service...) • Programming models based on middleware technologies (i.e. Java EE, .NET) • Service oriented architecture, Enterprise Service Bus (ESB)
Method and Extent of Examination	Written examination, 120 minutes
Pre-Exam Requirements	Assignments
Recommended Literature (Excerpt)	• Ayers, D., H. Bergsten et al.: Professional Java Server Programming. Wrox Press Ltd., 1999. • Brose, G., A. Vogel, K. Duddy: Java Programming with CORBA. John Wiley & Sons, 2001. • Burke, B., R. Monson-Haefel: Enterprise JavaBeans 3.0. O'Reilly, 2006. • Hall, M., L. Brown, Y. Chaikin: Core Servlets and JavaServerPages – Advance Technologies. Prentice Hall, 2007. • Hohpe, G., B. Woolf: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley, 2003. • Schalk, C., E. Burns: JavaServer Faces – The complete reference. Mc Graw Hill, 2007. • Singh, I., B. Stearns, M. Johnson: Designing Enterprise Applications with the J2EE Platform. Second Edition, Addison Wesley, 2002.

1.8 Software Project

Course	Master Software Technology
Name of Module	Software Project
Abbreviation	SOP
Semester	2
Responsible	Course Director Software Technology
Lecturers	All professors of computer science
Method of Teaching	Project with Lecture
European Credit Transfer Points	8 ECTS Points

Weekly Contact Hours	4h (4h Integrated Lecture/Exercises/Presentations)
Student Work Load	240h Total: • Project supervision (68h) • Project work and presentation (172h)
Necessary Previous Knowledge	Software Engineering, Software Project Management (Bachelor Level)
Final Knowledge and Skills	*Knowledge and understanding* On completion the student has knowledge and practical experience of software engineering while developing software in an industry-like project with real customers pertaining to software design, version control, documentation, testing, maintenance and software quality assurance. He or she has an understanding and experience of the difficulties of team management and troubleshooting (due to the size of the project team). *Disciplinary / professional skills* On completion the student is able to cope with typical situations arising in software projects by applying previously learned methods and team management.
Index	An industrial-like software project is supervised by faculty teachers who are usually playing the role of a customer. The software project team (6 – 10 students) participates in all stages of the software process from requirements engineering to roll-out. Beneath technical problem solving methods of management, leadership, planning, communication and cooperation have to be employed.
Method and Extent of Examination	Project work, documentation and presentation
Pre-Exam Requirements	None
Recommended Literature (Excerpt)	• Beck, K., M. Fowler: Planning extreme programming. Addison-Wesley, Boston, 2001 • Dustin, E., J. Rashka, J. Paul: Automated Software Testing: Introduction, Management, and Performance. Addison Wesley Publishing Company, 1999. • Haug, M., E.W. Olsen, G. Cuevas, S. Rementeria (Eds.): Managing the Change: Software Configuration and Change Management. Software Best Practice 2, Springer, 2001. • Kruchten, P.: The Rational Unified Process: An Introduction (2nd Edition). Addison-Wesley, 2000. • Robertson, S. J. Robertson: Mastering the Requirements Process. Addison Wesley, 1999.

1.9 Software Verification and Validation

Course	Master Software Technology
Name of Module	Software Verification and Validation
Abbreviation	SVV
Semester	2
Responsible	Prof. Dr. Heusch
Lecturers	Prof. Dr. Heusch, Prof. Dr. Homberger
Method of Teaching	Lecture with theoretical and practical exercises.
European Credit Transfer Points	3 ECTS Points
Weekly Contact Hours	2h (1h Lectures + 1h Exercises/Presentations)
Student Work Load	90h Total: • Lectures (34h) • Self Studies (56h)
Necessary Previous Knowledge	Software Engineering basics of Testing and Quality Assurance
Final Knowledge and Skills	Knowledge and understanding On completion the student knows about advanced techniques of software validation by using stochastic models and the formal methods for software verification based on first order logic and formal specification. Disciplinary / professional skills On completion the student is able to select an appropriate stochastic model and validate software. He or she can apply formal specification for program proofing and use Eiffel for creating provably verifiable software.
Index	• Requirements engineering as prerequisite for validation • Stochastic models of program behavior • First order logic and formal specification • Static analysis and program transformations • Use of model checking and deductive techniques • Using Eiffel for provably verifiable software
Method and Extent of Examination	Written examination, 90 minutes
Pre-Exam Requirements	Assignments
Recommended Litera-	• Meyer, B.: Object Oriented Software Construction. Prentice-Hall,

ture (Excerpt)	1997 • Wordsworth, J. B.: Software Development with Z - A Practical Approach to Formal Methods in Software Engineering. Addison-Wesley, 1992. • Zeller, A.: Why Programs fail: A guide to systematic debugging. dpunkt, 2005. • Various international research papers (distributed in class)

1.10 System Design

Course	Master Software Technology
Name of Module	System Design
Abbreviation	SYD
Semester	2
Responsible	Prof. Dr. Deininger
Lecturers	Prof. Dr. Deininger, Prof. Dr. Keller
Method of Teaching	Lecture with theoretical and practical exercises
European Credit Transfer Points	6 ECTS Points
Weekly Contact Hours	4h (3h Lectures + 1h Exercises)
Student Work Load	180h Total: • Lectures (68h) • Self Studies (112h)
Necessary Previous Knowledge	Software Engineering, Object Oriented Software Implementation
Final Knowledge and Skills	Knowledge and understanding On completion the student knows the different interrelationships between requirements and design and architectural choices of large-scale systems. He or she knows the principles of software design and the different design views and knows how a system design affects the testability of a system. Disciplinary / professional skills On completion the student is able to develop different design views and select fitting patterns for certain problems and draw from architectural choices, especially for large-scale systems. He or she is able to select and use appropriate modeling techniques. He or she can rate the consequences of certain design decisions
Index	• Basic principles of design: terms and definitions, abstraction, de-

	composition, decoupling. • Different design views and their elements. • Methods, notations and patterns for different design views • Measuring and testing of design. • Special Design Topics: Frameworks & Libraries, Persistence, User Interfaces
Method and Extent of Examination	Written examination, 120 minutes
Pre-Exam Requirements	Assignments
Recommended Literature (Excerpt)	• Bass, L., P. Clements, R. Kazman: Software Architecture in Practice, 3nd edition, Addison-Wesley Professional, 2012.. • Buschmann, F., R. Meunier, H. Rohnert, P. Sommerlad, M. Stal: Pattern-Oriented Software Architecture: A System of Patterns, John Wiley & Sons, 1996. • Clements, P., F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, J. Stafford: Documenting Software Architectures, Addison-Wesley , 2nd edition, Addison-Wesley, 2010. • Evans, E.: Domain-Driven-Design, Addison- Wesley, 2008. • Fowler, M.: Patterns of Enterprise Application Architecture; Addison- Wesley, 2014. • Gamma, E., R. Helm, R. Johnson, J. Vlissides: Design Patterns: Elements of Reusable OO Software. Addison-Wesley, 1997. • Meyer, B.: Object-Oriented Software Construction. Prentice Hall, 1997. • Shaw, M., P. Clements: The Golden Age of Software Architecture, IEEE SOFTWARE, March/April 2006, 31-39. • Szyperski, C.: Component Software - Beyond Object-Oriented Programming. Addison-Wesley, 2002. • Züllighoven, H.: Object-Oriented Construction Handbook: Developing Application-Oriented Software with the Tools & Materials Approach. Morgan Kaufmann, 2004.

1.11 Elective Module 2

Course	Master Software Technology
Name of Module	Elective Module 2
Abbreviation	EL2
Semester	2

1.11.1 Business Process Technologies

Course	Master Software Technology
Name of Module	Business Process Technologies
Abbreviation	BPT
Semester	2
Responsible	Prof. Dr. Höß
Lecturers	Prof. Dr. Höß, Prof. Dr. Kramer, Prof. Dr. Lückemeyer
Method of Teaching	Lecture with theoretical and practical exercises
European Credit Transfer Points	6 ECTS Points
Weekly Contact Hours	4h (3h Lectures + 1h Exercises/Presentations)
Student Work Load	180h Total: • Lectures (68h) • Self Studies (112h)
Necessary Previous Knowledge	Programming, Middleware Technology, Basic internet technology (XML, DTD, XML Schema, Namespaces, XPath, XSL, XSLT, DOM)
Final Knowledge and Skills	Knowledge and understanding On completion the student knows how to model inter- and intra-organizational business processes and technical workflows. He or she understands the concepts of orchestration of services and the choreography between services. He or she knows the necessary concepts and technologies for service-enabling existing legacy applications. Disciplinary / professional skills On completion the student is able to select and use modeling techniques for business processes and technical workflows and apply these techniques in a practical project. He or she can rate the technological and organizational implications when an IT landscape in a company has to be transformed into a process-oriented SOA.
Index	• Guided self-study for recapitulating basic internet technologies (XML, DTD, XML Schema, Namespaces XSLT, DOM) • Business Process Management (BPM) • Business Process Modeling with BPMN • Service-oriented Architectures and Web Services • (SOAP, WSDL, …) • Orchestration of services for execution (e.g. using BPEL) • Design and implementation of exemplary business processes which span across several IT systems • Case studies and practical examples

Method and Extent of Examination	Written examination, 90 minutes
Pre-Exam Requirements	Assignments, System exercises and presentations
Recommended Literature (Excerpt)	• Allweyer, T.: BPMN 2.0 – Introduction to the Standard for Business Process Modeling. Books on Demand, 2010. • Cummins, F.: Building the Agile Enterprise: With SOA, BPM and MBM. Morgan Kaufmann, 2009. • Krafzig, D., K. Banke, D. Slama: Enterprise SOA: Service Oriented Architecture Best Practices. Prentice Hall, 2005. • Margolis, B.: SOA – Concepts, BPEL and SCA. MC Press, 2007. • Weske, M.: Business Process Management – Concepts, Languages, Architectures. 2nd ed., Springer, 2012. • Weerawarana, S., F. Curbera, F. Leymann, T. Storey et al.: Web Services Platform Architecture. Prentice Hall, 2005. • Current research papers and online material on BPM, SOA, BPMN, BPEL and other topics

1.11.2 Business Intelligence

Course	Master Software Technology
Name of Module	Business Intelligence
Abbreviation	DWH
Semester	2
Responsible	Prof. Koch
Lecturers	Prof. Koch
Method of Teaching	Lecture with theoretical and practical exercises; independent group project with presentation
European Credit Transfer Points	6 ECTS Points
Weekly Contact Hours	4h (3h Lectures + 1h Exercises/Presentations)
Student Work Load	180h Total: • Lectures (68h) • Self Studies (112h)
Necessary Previous Knowledge	Database theory (especially normal forms, relational algebra, design procedures), relational systems, SQL, Middleware Technology, Bachelor-level mathematics
Final Knowledge and	*Knowledge and understanding*

Skills	On completion the student has a deeper understanding of goals and functionality of data warehouse systems. He or she has practical experience with a data warehouse system and insight into current business intelligence research issues. *Disciplinary / professional skills* On completion the student is able to evaluate strengths and weaknesses of data warehouse systems, to build a data warehouse system, and to make informed decisions about different situations of data warehouse usage in practical projects within enterprise contexts.
Index	• Purposes and application areas for data warehouses, case studies, comparison to database systems and transaction processing systems • Reference model for data warehouses, data acquisition, monitoring, extraction, transformation, loading, data marts versus data warehouse, data warehouse bus architecture • Data analysis: OLAP, data mining (statistical methods, regression, value prediction, decision trees, association discovery, a priori method, neural networks, visualization). • System architectures with middleware, web based architectures • Multidimensional models and algebra • Conceptual and physical modeling: multidimensional entity relationship model, schema evolution, star join schemas, snow flaking, array structures, performance optimization (materialized views, efficient indexing techniques) • Implementation of data warehouses with different DBMS types, ROLAP, MOLAP, HOLAP; OLAP extensions of SQL
Method and Extent of Examination	Written examination, 90 minutes
Pre-Exam Requirements	Successful seminar paper and presentation
Recommended Literature (Excerpt)	• Adamson, C., M. Venerable: Data Warehouse Design Solutions. Wiley, 1998. • Bauer, A., H. Günzel: Data Warehouse Systeme - Architektur, Entwicklung, Anwendung. dpunkt Verlag, 2008. • Kimball, R.: The Data Warehouse Toolkit - Practical Techniques for Building Dimensional Data Warehouses. Wiley, 1996. • Kimball, R., L. Reeves, M. Ross, W. Thornthwaite: The Data Warehouse Lifecycle Toolkit - Expert Methods for Designing, Developing, and Deploying Data Warehouses. Wiley, 1998. • Inmon, W.H.: Building the Data Warehouse. Wiley, 1996. • Humphries, M., M.W. Hawkins, M.C. Dy: Data Warehousing - Architecture and Implementation. Prentice Hall, 1999. • Course material, additional up-to-date articles available online in the Moodle System

1.11.3 Additional Elective Module 2

Course	Master Software Technology
Name of Module	Additional Elective Module 2
Abbreviation	ADD2
Semester	2
Responsible	Course Director of Software Technology
Lecturers	depending on actual topic
Method of Teaching	depending on actual topic
European Credit Transfer Points	6 ECTS Points
Weekly Contact Hours	4h
Student Work Load	180h Total: • Lectures (68h) • Self Studies (112h)
Necessary Previous Knowledge	depending on actual topic
Final Knowledge and Skills	depending on actual topic
Index	depending on actual topic
Method and Extent of Examination	to be defined by the examination board
Pre-Exam Require-ments	to be defined by the examination board
Recommended Litera-ture (Excerpt)	depending on actual topic

1.12 Master Thesis

Course	Master Software Technology
Name of Module	Master Thesis
Abbreviation	MT
Semester	3
Responsible	Supervising professor

Lecturers	-
Method of Teaching	Lecture with theoretical and practical exercises; independent group project with presentation
European Credit Transfer Points	30 ECTS Points
Weekly Contact Hours	-
Student Work Load	900h Total
Necessary Previous Knowledge	Modules of the previous semesters (at least 40 ECTS); the thesis topic has to be relevant to the taught modules of the master course.
Final Knowledge and Skills	On completion the student shows that he or she can solve in a predefined period a problem of his or her domain independently using scientific methods. This includes • sustained independent work of high quality fulfilling an agreed specification, • performing a critical review of research literature in the field of information technology, • analysis, synthesis and creative application of what has been learned in previous courses, • creation of a detailed and coherent report, in which the thesis work is presented in the context of the problem domain, with solutions proposed or implemented, justified and a critical appraisal of the work done.
Index	Depends on the actual domain specific topics; typically the thesis consists of one or more of the following activities: • Production of a detailed specification or design for a software system, or the implementation of one. Usually this includes a critical evaluation of the requirements and in the assessment of alternative tools, methods and solutions that could be employed and a conclusion which justifies the particular choices made. • Evaluation of some existing tools or technique or software system. Usually this includes the development and application of criteria in performing such an assessment. • Gathering of empirical evidence by directly testing existing tools or software system, and/or by seeking information from those who use (or would use in the case of a system to be developed) the system about aspects of its use. Usually this includes a justification of the approach taken in obtaining such evidence and in supporting the conclusions that can be drawn (or not drawn) from it.
Method and Extent of Examination	Written report, abstract, oral presentation
Pre-Exam Requirements	none
Recommended Litera-	depending on actual topic

ture (Excerpt)	