Module Handbook
Master Informatik / Computer Science (M-IN)

Faculty 2 - Technology, Computer Science and Economy

Full time studies Master Computer Science

Head of study course:
Prof. Dr. Marx
02.10.2024
Valid from winter term 2024/25 on

Table of Content

1 Compulsory Modules

1.1 Artificial Intelligence (M-IN-IN06)

Artificial Intelligence, Künstliche Intelligenz (KINT)						
ID	**Workload** 180h	**ECTS** 6	**Term at study start** ST (start): 2 WT (start): 1		**Frequency** winter term	**Duration** 1 term
1	**Course** Lecture plus workshops		**Contact time lecture** 30h	**Contact time other** 30h	**Self-studies** 120h	**Planned group size** 25 students
2	**Learning Outcomes** The students know advanced methods of artificial intelligence. Especially deep learning and deep reinforcement learning algorithms are understood by the students and can be applied to new problems. The students know how to train, tune and debug Deep Learning models.					
3	**Content** - Neuronal networks - Generative adversarial networks - Attacks against neuronal networks, adversarial examples - Convolutional neural networks - Recurrent neural networks - Reinforcement learning					
4	**Course form** Lecture combined with student workshops, project work and presentations; optional excursion					
5	**Prerequisites for attending** Formal: none Content: none					
6	**Form of examination** Assignment Project work and oral examination (assessment of the project presentation)					
7	**Prerequisites for granting ECTS** Presentation of assignment/project work with positive assessment					
8	**Utilization of the module (in other studies)** This module is not used in other courses					
9	**Weight for the final score** Weighting according to the ECTS points					
10	**Module commissioner:** Prof. Dr. Florian Dahms **Tutor:** Prof. Dr. Florian Dahms					
12	**Literature:** Stuart Russell, Peter Norvig; Artificial Intelligence: A Modern Approach, 4th Edition (2020) Ian Goodfellow, Yoshua Bengio, Aaron Courville; Deep Learning (2016) Richard Sutton, Andrew Barto; Reinforcement Learning: An Introduction (2018) C. Steger, M. Ulrich, C. Wiedemann: Machine Vision Algorithms and Applications, Wiley-VCH, ISBN 978-3-527-41365-2 F. Chollet: Deep Learning with Python, Manning Publications, ISBN 978-1617296864 https://docs.opencv.org/4.6.0/index.html https://pyimagesearch.com					

1.2 Architecture of Information Systems (M-IN-IN02)

Architecture of Information Systems, Architektur von Informationssystemen (SYSE)						
ID	**Workload** 180h	**ECTS** 6	**Term at study start** ST (start): 2 WT (start): 1		**Frequency** winter term	**Duration** 1 term
1	**Course** Lecture, Tutorial, Practical Project		**Contact time lecture** 30h	**Contact time other** 30h	**Self-studies** 120h	**Planned group size** 25 students
2	**Learning Outcomes** After successful completion of this module, students acquire the following competencies: - They are aware of additional challenges in the globalization of software development and can software development and can deal with them accordingly. - They can name and describe aspects that can influence motivation and productivity of software developers, including but not limited to psychological aspects - They are familiar with common architecture principles and can validate the validate compliance with design rules. - They have mastered simple DevOps techniques and are able to put software into operation in an automated and reproducible manner. - They can perform user interface testing as well as testing using mocks and mutants and automate these activities. - They are familiar with software maintenance challenges and can deal with them appropriately. - They can select a viable option for operation of large enterprise systems taking into account cloud, co-located and on-premise options. - They can apply Design by Contract to improve the safety of code. - They can monitor applications during test and production operations and propose suitable actions to solve arising issues. - They can evaluate and optimize processes in software-intensive environments.					
3	**Content** The course assumes previous knowledge in basic software engineering concepts and techniques as taught in Computer Science Bachelor programs. Building on this foundation, the course intensifies general understanding and practical actionability in the following areas: - Software Platforms - Cloud Computing - Global Software Engineering - Motivation and Productivity - Architecture Design Process and its Documentation - Architecture Validation, Acceptance Testing - System Introduction - Mock Testing - Mutant Testing and Evaluation of Unit Test Suites - Monitoring and Observability - Distribution, Cloud Computing - Operations - DevOps, Infrastructure as Code - Formal Methods and Design by Contract - Psychological Aspects and Dark Agile - Evaluation and Improvement using quality models (CMMI, Spice) - Model-Driven Architecture					

4	**Course form** Lecture, Tutorial, Practical Project
5	**Prerequisites for attending** Formal: none Content: none
6	**Form of examination** Written exam Presentation Term paper Oral examination Examination (successfully completed project, presentation and written paper)
7	**Prerequisites for granting ECTS** Passed exam plus study achievement
8	**Utilization of the module (in other studies)** This module is not used in other courses
9	**Weight for the final score** Weighting according to the ECTS points
10	**Module commissioner:** Prof. Dr. Daniel Kulesz **Tutor:** Prof. Dr. Daniel Kulesz
12	**Literature:** - Sommerville, Ian: Software Engineering. Pearson, 2018 - Sadowski, Caitlin, and Thomas Zimmermann: Rethinking productivity in software engineering, Springer Nature, 2019. - Le, D. "Na, Kumar, Rb, Nguyen, GN, Chatterjee, JMd: Cloud Computing and Virtualization, John Wiley and Sons, 2018 - Chaudhary, Mukund, and Abhishek Chopra: CMMI for Development, Springer 2017. - Ludewig, J. und Lichter, H.: Software Engineering - Grundlagen, Menschen, Prozesse Techniken, dpunkt, 4. Auflage, 2023 (German)

1.3 System Analysis (M-IN-IN04)

Systems Analysis, System Analyse (SYSA)						
ID	**Workload** 180h	**ECTS** 6	**Term at study start** ST (start): 2 WT (start): 1		**Frequency** winter term	**Duration** 1 term
1	**Course** Lecture and tutorials		**Contact time lecture** 30h	**Contact time other** 30h	**Self-studies** 120h	**Planned group size** 25 students
2	**Learning Outcomes** Students acquire knowledge of model building. They can classify and define systems and determine system boundaries. For the modeling and analysis of systems, students can use methods from different areas of computer science and mathematics.					
3	**Content** - Systems and models - Model building - Cellular automata - Learning agents - Chaos theory - Self-organizing systems - Game Theory - Swarm Intelligence - Stochastic processes and queues					
4	**Course form** Lecture and tutorials					
5	**Prerequisites for attending** Formal: none Content: none					
6	**Form of examination** presentation written exam					
7	**Prerequisites for granting ECTS** Passed exam					
8	**Utilization of the module (in other studies)** This module is not used in other courses					
9	**Weight for the final score** Weighting according to the ECTS points					
10	**Module commissioner:** Prof. Dr. Frank Mehler **Tutor:** Prof. Dr. Frank Mehler					
12	**Literature:** H. Bossel: Systeme, Dynamik, Simulation, Modellbildung. Analyse und Simulation komplexer Systeme, Norderstedt D. Imboden, S. Koch; Systemanalyse, Einführung in die mathematische Modellierung natürlicher Systeme, Springer-Verlag J. Schmidt, Ch. Klüver, J. Klüver: Programmierung naturanaloger Verfahren, Vie-weg+Teubner O. Loistl, Chaostheorie: Zur Theorie nichtlinearer dynamischer Systeme, Oldenbourg-Verlag Ch. Rieck, Spieltheorie, Eine Einführung, Eschborn Th. Schickinger, A. Steger, Diskrete Strukturen, Band 2: Wahrscheinlichkeitstheorie und Statistik					

1.4 Advanced Database Systems (M-IN-IN03)

Advanced Database Systems, Vertiefung Datenbanksysteme (VEDA)						
ID	**Workload** 180h	**ECTS** 6	**Term at study start** ST (start): 1 WT (start): 2		**Frequency** summer term	**Duration** 1 term
1	**Course** Lecture and Tutorials		**Contact time lecture** 60h	**Contact time other** 30h	**Self-studies** 90h	**Planned group size** 25 students
2	**Learning Outcomes** Students know the architecture and structure of relational database systems. They know physical storage and index structures. They understand the issues of multi-user synchronization, serializability even for long-running transactions, and logging and recovery. You understand the 2-phase commit protocol for distributed transactions. They know concepts of distributed database systems as well as for database replication. Students know the structure and tasks of a data warehouse. They know the meaning of ETL, different approaches to modeling the base database of a DWH (Inmon, Kimball, Data Vault) and the modeling of data cubes and data marts (Star Schema etc.). You are able to design a DWH and to implement its essential components exemplarily. You will be familiar with extended query options for a DWH, in particular using "Analytical SQL", and will be able to apply these in practice.					
3	**Content** - Layer models of database systems - Physical storage structures - Different index structures - Transaction management and advanced transaction concepts also for distributed databases - Database replication - Synchronization, locking procedures and serializability - Log files and recovery - Datawarehouse and OLAP: Architecture, Modeling, ETL, Analytical SQL					
4	**Course form** lecture and tutorials					
5	**Prerequisites for attending** Formal: none Content: Basics of database systems, especially relational databases					
6	**Form of examination** written exam oral examination the exam form is determined at the beginning of the semester					
7	**Prerequisites for granting ECTS** passed exam passed study achievement					
8	**Utilization of the module (in other studies)** This module is not used in other courses					
9	**Weight for the final score** Weighting according to the ECTS points					
10	**Module commissioner:** Prof. Dr. Michael Schmidt **Tutor:** Prof. Dr. Michael Schmidt					
12	**Literature:** - script of the lecture - Kemper, A.: „Datenbanksysteme“ , Oldenbourg, aktuelle Auflage					

	- Garcia-Molina, H..: „Database Systems - The Complete Book, Pearson - Heuer, A: „Datenbanken - Konzepte und Sprachen“, Mitp-Verlag - Heuer, A: „Datenbanken: Implementierungstechniken“, Mitp-Verlag - Hahne, M.:„Modellierung von Business Intelligence-Systemen, dpunkt.verlag - Kemper, H.G.: „Business Intelligence - Grundlagen und praktische Anwendungen", Vie-weg+Teubner - Köppen v. et al.: „Data Warehouse Technologien" - Lehner W.: „Datenbanktechnologie für DWH-Systeme", dpunkt.verlag - Bauer A. et al.: „Data Warehouse Systeme", dpunkt.verlag

1.5 Scientific Seminar (M-IN-IN05)

Scientific Course, Wissenschaftliches Seminar (WISE)						
ID	**Workload** 180h	**ECTS** 6	**Term at study start** ST (start): 2 WT (start): 1		**Frequency** summer term	**Duration** 1 term
1	**Course** Seminar		**Contact time lecture** 30h	**Contact time other** 30h	**Self-studies** 120h	**Planned group size** 25 students
2	**Learning Outcomes** Students are able to compile the state of the art of a specific research topic in the field of computer science as well as to understand the content of a scientific paper. They are able to put together a scientifc presentation plus to give a lecture on it. The students have the ability to classify and evaluate a scientific contribution and to differentiate between its significance for research and application. Furthermore, the students have acquired in-depth knowledge and skills for scientific work.					
3	**Content** - Up to date /latest scientific publications from different areas fo Computer Science, like database technologies, cybersecurity, robotics, system architectures, software-engineering, artificial intelligence, operating systems, post-quantum cryptography, web technologies, mobile systems etc.					
4	**Course form** seminaristic					
5	**Prerequisites for attending** Formal: none Content: none					
6	**Form of examination** Oral examination, presentation; The form of examination will be determined at the beginning of the course. English lecture, min. 60 min.					
7	**Prerequisites for granting ECTS** Passed exam					
8	**Utilization of the module (in other studies)** This module is not used in other courses					
9	**Weight for the final score** Weighting according to the ECTS points					
10	**Module commissioner:** Prof. Dr. Thomas Marx **Tutor:** Prof. Dr. Thomas Marx					
12	**Literature:** Current scientific papers of the recent 1-2 years, accepted (blind referee) at scientific conferences (e.g. published Lecture Notes in Computer Science etc.)					

1.6 Higher Mathematics (M-IN-MNS01)

<table>
<tr><td colspan="7"><b>Higher Mathematics for Information Systems, Höhere Mathematik (HÖMA)</b></td></tr>
<tr><td><b>ID</b></td><td><b>Workload</b>
180h</td><td><b>ECTS</b>
6</td><td colspan="2"><b>Term at study start</b>
ST (start): 1
WT (start): 2</td><td><b>Frequency</b>
summer term</td><td><b>Duration</b>
1 term</td></tr>
<tr><td>1</td><td colspan="2"><b>Course</b>
Lecture and tutorials</td><td><b>Contact time lecture</b>
30h</td><td><b>Contact time other</b>
30h</td><td><b>Self-studies</b>
120h</td><td><b>Planned group size</b>
25 students</td></tr>
<tr><td>2</td><td colspan="6"><b>Learning Outcomes</b>
Students know the basic concepts, theorems and algorithms of algebra and discrete mathematics, which are essential for a deeper understanding of various areas of
- theoretical computer science (such as algorithms, data structures, languages and complexity theory) and
- applied computer science (such as cryptography and coding theory) are needed. You will be able to apply these concepts and algorithms.
They know the basic concepts of a structure-oriented algebra such as substructure, factor structure, homo- and isomorphism. They know basic notions of order theory and elementary examples of partially ordered sets.
The students deepen their abilities to understand formal arguments and to formulate them themselves in a technically precise way with regard to a possible own scientific activity.</td></tr>
<tr><td>3</td><td colspan="6"><b>Content</b>
- Relations (equivalence, order, congruence relations)
- semigroups, monoids, groups, rings, solids
- Group theory (subgroup, normal divisor, factor group, homomorphism theorem)
- Representation of groups with generators and relations, with permutations and with matrices
- Ordered sets (general terms and constructions, as well as standard examples from combinatorics)</td></tr>
<tr><td>4</td><td colspan="6"><b>Course form</b>
Lecture and tutorials</td></tr>
<tr><td>5</td><td colspan="6"><b>Prerequisites for attending</b>
Formal: none
Content: none</td></tr>
<tr><td>6</td><td colspan="6"><b>Form of examination</b>
Written exam</td></tr>
<tr><td>7</td><td colspan="6"><b>Prerequisites for granting ECTS</b>
Passed exam</td></tr>
<tr><td>8</td><td colspan="6"><b>Utilization of the module (in other studies)</b>
This module is not used in other courses</td></tr>
<tr><td>9</td><td colspan="6"><b>Weight for the final score</b>
Weighting according to the ECTS points</td></tr>
<tr><td>10</td><td colspan="6"><b>Module commissioner:</b> Prof. Dr. Tino Schürg
<b>Tutor:</b> Prof. Dr. Tino Schürg</td></tr>
<tr><td>12</td><td colspan="6"><b>Literature:</b>
- Fraleigh: A First Course in Abstract Algebra
- Pinter - A Book of Abstract Algebra 2nd ed.
- Witt: Algebraische und zahlentheoretische Grundlagen der Informatik (eBook) - Davey , Priestley: Introduction to Lattices and Order, 2nd ed.
- Ganter: Diskrete Mathematik: Geordnete Mengen</td></tr>
</table>

1.7 Master Thesis including Colloquium (M-IN-PP01)

Master Thesis including Colloquium, Masterarbeit inclusive Kolloquium (MAST)						
ID	**Workload** 450h	**ECTS** 30	**Term at study start** ST (start): 3 WT (start): 3		**Frequency** any term	**Duration** 1 term
1	**Course** Self-study and consultations		**Contact time lecture** 0h	**Contact time other** 0h	**Self-studies** 450h	**Planned group size** 25 students
2	**Learning Outcomes** Students are enabled to independently solve a complex problem or task from science, industry or society. or society independently and solve them. They are able to assess and evaluate different assess and evaluate different solution approaches. To solve the problem, they apply the technical knowledge acquired during their studies. Students plan and organize their academic organize their scientific work independently. They can analyze and evaluate scientific sources of information. be analyzed and evaluated. The results are formulated and presented with scientific precision in the Master's thesis. In the colloquium, students present their approach, methods and results coherently and logically.					
3	**Content** The Master thesis is written either at the university or at or in cooperation with a company/institution. company/institution. The university lecturer acts as supervisor. He or she supports the students in personal discussions with regards to compliance with the above-mentioned learning and qualification objectives. Depending on the task, several students can also work on the same project, each of them independently.					
4	**Course form** Coaching, individual consultations, colloquium					
5	**Prerequisites for attending** Formal: none Content: none					
6	**Form of examination** Thesis and Presentation/Colloquium (max. 30 minutes)					
7	**Prerequisites for granting ECTS** Passed exam (thesis and colloquium)					
8	**Utilization of the module (in other studies)** This module is not used in other courses					
9	**Weight for the final score** Weighting according to the ECTS points					
10	**Module commissioner:** Prof. Dr. Thomas Marx **Tutor:** all computer science professors					
12	**Literature:** Sample master's theses and presentations for the colloquium as well as a list of recommended basic literature are provided on the Internet					

2 Complementary Modules (Computer Science)

2.1 Advanced Data Mining with R and JavaScript on GNU/Linux (M-IN-WP-36) / AI

Advanced Data Mining with R and JavaScript on GNU/Linux, Fortgeschrittenes Data Mining mit R und JavaScript auf GNU/Linux (ADAM)						
ID	**Workload** 180h	**ECTS** 6	**Term at study start** ST (start): 2 WT (start): 1		**Frequency** winter term	**Duration** 1 term
1	**Course** Lecture and Tutorial		**Contact time lecture** 30h	**Contact time other** 30h	**Self-studies** 120h	**Planned group size** 25 students
2	**Learning Outcomes** After completing the module, students will be able to: - apply basic empiric methods to evaluate characteristics of experimental data - classify and apply basic methods and algorithms of data mining for the analysis of scientific data - to write small programs independently in the statistical programming language R and JavaScript - generate reusable data analysis and visualization web components using HTML, CSS, and Javascript - use such web components to analyze and visualize data - create interactive scientific plots that enable the user to better explore scientific data and thus aid the scientist in hypothesis formation and validation					
3	**Content** NOTE that the WHOLE COURSE will be held on a GNU/Linux operating system. Students are highly recommended to prepare their hardware either with a (dual boot) GNU/Linux operating system or use a virtual machine GNU/Linux installation, or (least recommended) use the Windows 10 or 11 subsystem. The course covers the following topics - Introduction to or repetition of basic statistics, respectively - standard algorithms and methods in applied data science, and implementation in R and/or JavaScript; these comprise: - statistical distribution function estimation methods - normalization and data transformation - Distances and correlation coefficients - Clustering and classification, basics of data mining - Regression and basic statistical learning methods - principal component analysis - basics in text mining and text corpus analysis - Visualization of results (boxplot, heat map, dendrogram, etc.) Additional topics are: - Basics of the statistical programming language R - Basics of ECMAScript and its usage in statistics and web component development - programming of reusable web components: covering specifics like the shadow DOM and asynchronous functions					
4	**Course form** Lecture and Tutorial					
5	**Prerequisites for attending** Formal: none Content: In order to be able to pass the course successfully, students must have basic experience in programming of JavaScript and some other programming language. Student must be able to use the GNU/Linux operating system, particularly the terminal and command-line-interface. Students must have experience in using code versioning tools like git.					
6	**Form of examination** Written exam Project Work					

	Oral Exam Presentation
7	**Prerequisites for granting ECTS** passed examination passed study achievement Explanations: Passed module examination (examination performance)
8	**Utilization of the module (in other studies)** This module is not used in other courses.
9	**Weight for the final score** Weighting according to the ECTS points
10	**Module commissioner:** Prof. Dr. Asis Halab **Tutor:** Prof. Dr. Asis Halab
12	**Literature:** - https://www.w3schools.com/r/default.asp - https://www.w3schools.com/js/default.asp - James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer. - https://en.wikibooks.org/wiki/Statistics - Heumann, C., & Shalabh, M. S. (2016). Introduction to statistics and data analysis. Springer International Publishing Switzerland.

2.2 Computer Vision (M-IN-WP-35) / AI

<table>
<tr><td colspan="7"><b>Computer Vision, aktives Sehen (COVI)</b></td></tr>
<tr><td><b>ID</b></td><td><b>Workload</b>
180h</td><td><b>ECTS</b>
6</td><td colspan="2"><b>Term at study start</b>
ST (start): 2
WT (start): 1</td><td><b>Frequency</b>
winter term</td><td><b>Duration</b>
1 term</td></tr>
<tr><td>1</td><td colspan="2"><b>Course</b>
Lecture
Workshop
Tutorial
Excursion (optional)</td><td><b>Contact time lecture</b>
30h</td><td><b>Contact time other</b>
30h</td><td><b>Self-studies</b>
120h</td><td><b>Planned group size</b>
25 students</td></tr>
<tr><td>2</td><td colspan="6"><b>Learning Outcomes</b>
The students learn the complete process chain of computer vision from image acquisition and data transfer to computational image analysis. They are familiar with the most important machine vision algorithms and are practiced in the application of free open-source software (OpenCV and Keras/TensorFlow with Python-API) and proprietary software (e.g. HALCON or VisionPro).
The different approaches and pros/cons of traditional image processing versus deep learning techniques are understood.
The students are able to familiarize themselves with new topics in the field of computer vision and can present their acquired knowledge in an understandable way.</td></tr>
<tr><td>3</td><td colspan="6"><b>Content</b>
- Introduction and Overview
- Image Acquisition (illumination, lenses, cameras, data interfaces)
- Machine Vision Algorithms (data stuctures, image enhancement, geometric transformations, image segmentation, feature extraction, morphology, edge extraction, camera calibration, 3D-reconstruction, optical character recognition)
- Deep Learning for Machine Vision
- Machine Vision Applications with OpenCV, Keras/TensorFlow and HALCON or VisionPro
Optional (if possible): Excursion to a company in the field of Computer Vision</td></tr>
<tr><td>4</td><td colspan="6"><b>Course form</b>
Lecture combined with student workshops, project work and presentations; optional excursion</td></tr>
<tr><td>5</td><td colspan="6"><b>Prerequisites for attending</b>
Formal: none
Content: none</td></tr>
<tr><td>6</td><td colspan="6"><b>Form of examination</b>
Assignment
Project work and oral examination (assessment of the project presentation)</td></tr>
<tr><td>7</td><td colspan="6"><b>Prerequisites for granting ECTS</b>
Presentation of assignment/project work with positive assessment</td></tr>
<tr><td>8</td><td colspan="6"><b>Utilization of the module (in other studies)</b>
This module is not used in other courses</td></tr>
<tr><td>9</td><td colspan="6"><b>Weight for the final score</b>
Weighting according to the ECTS points</td></tr>
<tr><td>10</td><td colspan="6"><b>Module commissioner:</b> Dipl.-Phys. Michael Haag-Pichl
<b>Tutor:</b> Dipl.-Phys. Michael Haag-Pichl</td></tr>
<tr><td>12</td><td colspan="6"><b>Literature:</b>
A. Nischwitz, M. Fischer, P. Haberäcker, G. Socher: Bildverarbeitung,
Springer Vieweg, ISBN 978-3-658-28704-7
C. Steger, M. Ulrich, C. Wiedemann: Machine Vision Algorithms and Applications,
Wiley-VCH, ISBN 978-3-527-41365-2
F. Chollet: Deep Learning with Python, Manning Publications, ISBN 978-1617296864
https://docs.opencv.org/4.6.0/index.html
https://pyimagesearch.com</td></tr>
</table>

2.3 Planning and Scheduling (M-IN-WP-32) / AI

<table>
<tr><td colspan="7"><b>Planning and Scheduling, Planen und Terminisierung (PLANS)</b></td></tr>
<tr><td><b>ID</b></td><td><b>Workload</b>
180h</td><td><b>ECTS</b>
6</td><td colspan="2"><b>Term at study start</b>
ST (start): 1
WT (start): 2</td><td><b>Frequency</b>
summer term</td><td><b>Duration</b>
1 term</td></tr>
<tr><td>1</td><td colspan="2"><b>Course</b>
Lecture and Tutorials</td><td><b>Contact time lecture</b>
30h</td><td><b>Contact time other</b>
30h</td><td><b>Self-studies</b>
120h</td><td><b>Planned group size</b>
25 students</td></tr>
<tr><td>2</td><td colspan="6"><b>Learning Outcomes</b>
The students know concepts, methods, and tools for task-level planning and scheduling.
Methodological competency: The students know the state-of-the-art in task planning and are able to select and apply adequate methods for use in robotics applications.
Individual competency: Improved ability to perform abstract thinking and logical reasoning. Ability to formalize domain concepts in appropriate logics.</td></tr>
<tr><td>3</td><td colspan="6"><b>Content</b>
• Knowledge representation
• Formalizing action and action theories
• State space planning: STRIPS and friends
• Plan space planning: POP and friends
• Graph-based planning
• SAT-based planning
• HTN planning
• Scheduling and resource constraints
• Conditional planning, POMDPs
• Planning for multiagent systems</td></tr>
<tr><td>4</td><td colspan="6"><b>Course form</b>
• Attendance study: Presents lecturers, moderated discussions, group work
• Online supervision: (digital) exercises, repetitions (individual or in groups), in-depth studies (quantitative and qualitative methods)
• Self-study: learning with study letters, source study, exercises for self-study.</td></tr>
<tr><td>5</td><td colspan="6"><b>Prerequisites for attending</b>
Formal: none
Content: none</td></tr>
<tr><td>6</td><td colspan="6"><b>Form of examination</b>
project work
oral examination</td></tr>
<tr><td>7</td><td colspan="6"><b>Prerequisites for granting ECTS</b>
Passed exam</td></tr>
<tr><td>8</td><td colspan="6"><b>Utilization of the module (in other studies)</b>
This module is not used in other courses</td></tr>
<tr><td>9</td><td colspan="6"><b>Weight for the final score</b>
Weighting according to the ECTS points</td></tr>
<tr><td>10</td><td colspan="6"><b>Module commissioner:</b> Prof. Dr. Thomas Marx
<b>Tutor:</b> Iman Awaad (MSc Computer Science)</td></tr>
<tr><td>12</td><td colspan="6"><b>Literature:</b>
- Ghallab & Nau & Traverso: Automated Planning. Morgan Kaufmann, 2004.
- Russell & Norvig: Artificial Intelligence – A Modern Approach, 3rd edition. Prentice Hall, 2002.
- Richard Conway, William Maxwell, Louis Miller: Theory of Scheduling, Dover Publications, 1967
- Dana S. Nau. 2007. Current trends in automated planning. AI Magazine, Vol. 28, No. 4.</td></tr>
</table>

	- Dana S. Nau, Malik Ghallab, and Paolo Traverso. 2015. Blended planning and acting: preliminary approach, research challenges. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15). AAAI Press 4047-4051

2.4 Network Security (M-IN-WP-41)

Network Security, Netzwerksicherheit (NETS)						
ID	**Workload** 180h	**ECTS** 6	**Term at study start** ST (start): 1 WT (start):2		**Frequency** summer term	**Duration** 1 term
1	**Course** Lecture and Tutorials		**Contact time lecture** 30h	**Contact time other** 30h	**Self-studies** 120h	**Planned group size** 25 students
2	**Learning Outcomes** Network security is a critical component of modern information technology systems. After attending this course, the students will be able to: - describe different network architectures and concepts and be able to evaluate them with regards to their security properties, - reproduce which different typical threats exist in the network and which challenges exist, - analyze and evaluate a given network architecture with modern tools, such as nmap and wireshark, - know and apply different strategies and tools for detection and response and evaluate them in terms of advantages and disadvantages, - know and be able to apply security measures and protocols on the different network layers, - use classic network security tools such as firewalls and intrusion detection systems, including their placement in the network topology, - develop suitable response strategies and to solve security problems of other exemplary topics such as in the wireless networking or distributed systems domain.					
3	**Content** Attacks and defenses in the context of network and operating system security, including: - Principles of networking fundamentals and IT security concepts - Vulnerability and risk assessment using scanning tools like Nmap and monitoring tools such as Wireshark - Attacks and security measures for different network layers, including application layer (PGP, S/Mime, Web security, DNSSEC), Transport layer (TLS), network layer (IPSEC), data link (PPPoE) and medium access layer (WPA*). - Secure networking architecture elements such as firewalls, intrusion detection systems, monitoring systems, virtual private networks - Security measures for authentication, anonymity, and trust, especially in distributed systems - Network steganography - Decentralized systems: Blockchain, peer-to-peer networks, opportunistic networks.					
4	**Course form** Lecture and Tutorials					
5	**Prerequisites for attending** Formal: none Content: IT Security, Communication Networks					
6	**Form of examination** Written examination					
7	**Prerequisites for granting ECTS** Passed exam					
8	**Utilization of the module (in other studies)** This module is not used in other courses					

9	**Weight for the final score** Weighting according to the ECTS points
10	**Module commissioner:** Prof. Dr. Jens Reinhardt **Tutor:** Prof. Dr. Kálmán Graffi
12	**Literature:** - James F. Kurose and Keith W. Ross: "Computer Networking: A Top-Down Approach" - Charlie Kaufman, Radia Perlman, and Mike Speciner: "Network Security: Private Communication in a Public World" - Ross Anderson: "Security Engineering: A Guide to Building Dependable Distributed Systems" Steffen Wendzel:” IT-Sicherheit für TCP/IP- und IoT-Netzwerke: Grundlagen, Konzepte, Protokolle, Härtung (German Edition)

2.5 ERP in the cloud (M-IN-WP40)

ERP in the Cloud, ERP in der Cloud (ERPC)						
ID	**Workload** 180h	**ECTS** 6	**Term at study start** ST (start): 1 WT (start): 2		**Frequency** winter term	**Duration** 1 term
1	**Course** Lecture plus workshops		**Contact time lecture** 30h	**Contact time other** 30h	**Self-studies** 120h	**Planned group size** 25 students
2	**Learning Outcomes** The students know basic principles about Cloud Computing, Cloud Development, SAP Business Technology Platform and how international companies make use of SAP Business Technology Platform and other Cloud Products. Especially the required techniques and programming languages for Cloud Development are understood by the students. The students know how to develop, deploy, test and run Cloud Application on SAP BTP.					
3	**Content** - GxP requirements, Documentation Practices (optional) - Cloud Computing, IaaS, PaaS, SaaS - Business Technology Platform (BTP) Account Structure, Services, Integration in existing landscape - BTP BAS (Business Aplication Studio), Good coding principles - Security/Authentication/Authorization in the cloud - UI5 Workframe, CAP Modell, ODATA Protocol, CDS (HDI Container) - LC/NC (low code/no code) Development (controls/navigation)					
4	**Course form** Lecture combined with student workshops, project work					
5	**Prerequisites for attending** Formal: none Content: JavaScript, Web technologies (e.g HTML, CSS, etc), APIs, CRUD Operations					
6	**Form of examination** Project work and oral examination (assessment of the project presentation)					
7	**Prerequisites for granting ECTS** Presentation of assignment/project work with positive assessment					
8	**Utilization of the module (in other studies)** This module is not used in other courses					
9	**Weight for the final score** Weighting according to the ECTS points					
10	**Module commissioner:** Sven-Gerrit Dieckmann **Tutor:** Sven-Gerrit Dieckmann					
12	**Literature:** SAP Academy https://open.sap.com/ UI5 Documentation https://ui5.sap.com CAP Dokumentation https://cap.cloud.sap/docs/about/ UI5 Walkthrough https://ui5.sap.com/#/topic/3da5f4be63264db99f2e5b04c5e853db OData Documentation https://www.odata.org/documentation/					

2.6 Natural Language Processing (M-IN-WP-34) / AI

<table>
<tr><td colspan="7"><b>Natural Language Processing, Verarbeitung natürlicher Sprache</b> (NALP)</td></tr>
<tr><td><b>ID</b></td><td><b>Workload</b>
180h</td><td><b>ECTS</b>
6</td><td colspan="2"><b>Term at study start</b>
ST (start): 1
WT (start): 2</td><td><b>Frequency</b>
winter term</td><td><b>Duration</b>
1 term</td></tr>
<tr><td>1</td><td colspan="2"><b>Course</b>
Lecture plus workshops</td><td><b>Contact time lecture</b>
30h</td><td><b>Contact time other</b>
30h</td><td><b>Self-studies</b>
120h</td><td><b>Planned group size</b>
25 students</td></tr>
<tr><td>2</td><td colspan="6"><b>Learning Outcomes</b>
Students learn the fundamentals of automatically processing natural language. They know how to turn sentences into features and how machine learning models can be trained and applied to them. The students know how to solve common applications like sentiment analysis, translation, speech recognition and speech synthesis. They are familiar with common frameworks for implementing natural language processing systems.</td></tr>
<tr><td>3</td><td colspan="6"><b>Content</b>
- Tokenization, stemming, chunking
- Word embeddings
- Recurrent neural networks
- Attention mechanisms and transformers
- Sentiment analysis
- Machine translation
- Speech recognition and synthesis
- Ethical aspects of natural language generation</td></tr>
<tr><td>4</td><td colspan="6"><b>Course form</b>
Lecture combined with student workshops, project work</td></tr>
<tr><td>5</td><td colspan="6"><b>Prerequisites for attending</b>
Formal: none
Content: none</td></tr>
<tr><td>6</td><td colspan="6"><b>Form of examination</b>
Oral examination, presentation or written exam</td></tr>
<tr><td>7</td><td colspan="6"><b>Prerequisites for granting ECTS</b>
Passed exam</td></tr>
<tr><td>8</td><td colspan="6"><b>Utilization of the module (in other studies)</b>
This module is not used in other courses</td></tr>
<tr><td>9</td><td colspan="6"><b>Weight for the final score</b>
Weighting according to the ECTS points</td></tr>
<tr><td>10</td><td colspan="6"><b>Module commissioner:</b> Prof. Dr. Florian Dahms
<b>Tutor:</b> Prof. Dr. Florian Dahms</td></tr>
<tr><td>12</td><td colspan="6"><b>Literature:</b>
Current publications in the field of natural language processing</td></tr>
</table>

2.7 Autonomous and Mobile Robots (M-IN-WP-33) / AI

Autonomous and Mobile Robots, autonome und mobile Roboter (AROB)						
ID	**Workload** 180h	**ECTS** 6	**Term at study start** ST (start): 1 WT (start): 2		**Frequency** winter term	**Duration** 1 term
1	**Course** Lecture plus workshops		**Contact time lecture** 30h	**Contact time other** 30h	**Self-studies** 120h	**Planned group size** 25 students
2	**Learning Outcomes** - Students will be able to describe and classify the different AI paradigms for mobile robots (reactive, deliberative, hybrid). - Students can explain and evaluate the most important sensors and actuators for mobile robots. Students can describe compare and use the basic planning and navigation methods in mobile robotics. - Students can discuss basic approaches to robot learning and multi-robot and human-robot interaction. - Students can present the state of knowledge and current trends in mobile robotics and explain them using example robots. - Students will be able to design and program mobile robots yourself.					
3	**Content** - Reactive behavior - Sensors - Actuators, kinematics of drives - Hybrid deliberative/reactive behavior - Action planning - maps, self-localization - path planning, navigation - Robot learning - Error detection and healing - Multi-robot - Human-robot interaction - Current trends - example platforms					
4	**Course form** Lecture combined with student workshops, project work					
5	**Prerequisites for attending** Formal: none Content: none					
6	**Form of examination** **Oral exam, presentation, project work**					
7	**Prerequisites for granting ECTS** Passed exam					
8	**Utilization of the module (in other studies)** This module is not used in other courses					
9	**Weight for the final score** Weighting according to the ECTS points					
10	**Module commissioner:** Prof. Dr. Thomas Marx **Tutor:** Prof. Dr. Thomas Marx					
12	**Literature:** - Siciliano, Bruno; Khatib, Oussama: Handbook of Robotics. Springer. Berlin-Heidelberg. 2016. - J. Hertzberg, K. Lingemann, A. Nüchter: Mobile Roboter - Springer Vieweg 2012					

	- R. Siegwart, I. R. Nourbakhsh: Introduction to Autonomous Mobile Robots - Cambridge, MA: The MIT Press 2011 - R. R. Murphy: Introduction to AI Robotics - Cambridge, MA: The MIT Press 2000

2.8 Individual Profiling (M-IN-WP-28)

Individual Profiling (IPROF)						
ID	**Workload** 180h	**ECTS** 6	**Term at study start** ST (start): 1 WT (start): 2		**Frequency** any term	**Duration** 1 term
1	**Course** Self-studies and consultations		**Contact time lecture** 0h	**Contact time other** 30h	**Self-studies** 150h	**Planned group size** 25 students
2	**Learning Outcomes** The elective aims at the individual profile formation of the students. Within the framework of a freely defined task that they can solve complex problems largely independently with limited support from the supervisor to a large extent independently. It is expected that the students independently familiarize themselves with the necessary techniques for solving the problem posed. The problems to be worked on should be posed in such a way that they cannot be solved completely by means of compulsory lectures.					
3	**Content** The content forms current areas of computer science in which students wish to delve. The choice of the topic takes place in dialogue between students and university lecturer.					
4	**Course form** 2 SWS consultations					
5	**Prerequisites for attending** Formal: none Content: none					
6	**Form of examination** Presentation Term Paper					
7	**Prerequisites for granting ECTS** Passed exam					
8	**Utilization of the module (in other studies)** This module is not used in other courses					
9	**Weight for the final score** Weighting according to the ECTS points					
10	**Module commissioner:** Prof. Dr. Thomas Marx **Tutor:** Computer Science Professor at TH-Bingen					
12	**Literature:** Current literature depending on the chosen topic.					

2.9 New Database Systems (M-IN-WP-22)

<table>
<tr><td colspan="7">**New Database Systems, Neue Datenbanksysteme (NDBS)**</td></tr>
<tr><td>**ID**</td><td>**Workload**
180h</td><td>**ECTS**
6</td><td colspan="2">**Term at study start**
ST (start): 1
WT (start): 2</td><td>**Frequency**
summer term</td><td>**Duration**
1 term</td></tr>
<tr><td>1</td><td colspan="2">**Course**
Lecture plus tutorials</td><td>**Contact time lecture**
30h</td><td>**Contact time other**
30h</td><td>**Self-studies**
120h</td><td>**Planned group size**
25 students</td></tr>
<tr><td>2</td><td colspan="6">**Learning Outcomes**
The students know the theoretical basics of NoSQL database systems. They are familiar with the concepts of key vault stores, wide column stores, graph databases and document stores and can assess in which scenarios these database technologies can be used sensibly.
Object-relational mapping technologies (especially JPA) are known and can be applied in own applications.
Object-relational extensions of relational databases are known and can be used in examples. The students know basic concepts of OODBMS.
The interaction of XML and relational databases is known (SQL/XML) and can be used for generating XML documents from relational structures as well as for querying XML documents in the database using XQuery.
The interaction of JSON and relational databases is known (SQL/JSON) and can be applied for the generation of JSON documents from relational structures as well as for the query of JSON documents in the database.
The students know basic concepts of and application fields for "in-memory databases".
Emphasis and exact contents will be agreed upon at the beginning of the course, whereby also current developments in the area of DBMS will be considered. The learning and qualification objectives will be adjusted accordingly, if necessary.</td></tr>
<tr><td>3</td><td colspan="6">**Content**
- Basics of NoSQL databases (CAP theorem, BASE, Consistent Hashing, Map-Reduce, etc.)
- Types of NoSQL databases (key vault stores, wide column stores, graph databases, document stores)
- Object Relational Mapping with JPA
- OODBMS and ORDBMS
- SQL/XML incl. XQuery
- SQL/JSON
- In-Memory DBMS</td></tr>
<tr><td>4</td><td colspan="6">**Course form**
Lecture combined with student workshops, project work</td></tr>
<tr><td>5</td><td colspan="6">**Prerequisites for attending**
Formal: none
Content: Module “Database Systems” of Bachelor in Computer Science</td></tr>
<tr><td>6</td><td colspan="6">**Form of examination**
Written exam
Presentation
Term paper
Oral exam
Preferably oral examination or lecture</td></tr>
<tr><td>7</td><td colspan="6">**Prerequisites for granting ECTS**
Passed exam</td></tr>
</table>

8	**Utilization of the module (in other studies)** This module is not used in other courses
9	**Weight for the final score** Weighting according to the ECTS points
10	**Module commissioner:** Prof. Dr. Michael Schmidt **Tutor:** Prof. Dr. Michael Schmidt
12	**Literature:** Kemper, A.: „Datenbanksysteme“, aktuelle Auflage, Oldenbourg - Müller, B.; Wehr, H.: „Java Persitence API 2“, Hanser - Edlich et al.: NoSQL - Einstieg in die Welt nichtrelationaler WEB 2.0 Datenbanken, Hanser - Plattner H.; Zeier A.: „In-Memory Data Management“, Springer - Plattner H.: „Lehrbuch In-Memory Data Management: Grundlagen der In-Memory-Technologie“, Springer - Meier A., Kaufmann M.: "SQL- & NoSQL-Datenbanken", 2016 Springer, eBook - Lehner W.;Schöning H.: „XQuery – Grundlagen und fortgeschrittene Methoden“, dpunkt.verlag - weitere Literatur je nach Schwerpunkten - Fasel D., Meier A.: "Big Data - Grundlagen, Systeme und Nutzungspotenziale", 2016, Springer, eBook

2.10 Simulation (M-IN-WP-09)

Simulation, Simulation (SIMU)						
ID	**Workload** 180h	**ECTS** 6	**Term at study start** ST (start): 1 WT (start): 2		**Frequency** summer term	**Duration** 1 term
1	**Course** Lecture plus tutorials		**Contact time lecture** 30h	**Contact time other** 30h	**Self-studies** 120h	**Planned group size** 25 students
2	**Learning Outcomes** The students know the methodical basics of modeling and simulation of systems from diverse various application areas. They are familiar with the most important components, the mode of operation and the handling of a simulation system. The students know the different methods of time control. They are able to understand and deal with simulation languages and systems. Furthermore, the students are able to independently develop a model for a concrete problem, to implement it and to use it for simulation. To develop and implement a model for a concrete problem and to carry out simulations professionally. In addition, you will be able to independently develop software components of a simulation system or individually adapt existing.					
3	**Content** - Problem of modeling and simulation - Concepts of model building - Continuous models: methods for obtaining the system equations in different application areas - Methods of continuous simulation (numerical procedures for the solution of the occurring equations) - Discrete models (decision models, sequence problems, events) - Methods of discrete simulation (Petri nets, cellular automata, scheduling) - Simulation systems/simulators (presentation of different systems and their use) - simulation languages - Analysis and interpretation of simulation experiments - Validation and verification of a simulation model by implementation in a simulation system.					
4	**Course form** Lecture combined with tutorials (2 SWS each)					
5	**Prerequisites for attending** Formal: none Content: High school Mathematics					
6	**Form of examination** Oral or written exam					
7	**Prerequisites for granting ECTS** Passed exam					
8	**Utilization of the module (in other studies)** This module is not used in other courses					
9	**Weight for the final score** Weighting according to the ECTS points					
10	**Module commissioner:** Text **Tutor:** Prof. Dr.-Ing. Luckas					
12	**Literature**: J. Banks (ed.): Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice: Modelling, Estimation and Control. John Wiley & Sons, ISBN 978-0-471-13403-9 J. Banks, J. S. II Carson, B. L. Nelson, D. M. Nicol: Discrete-Event System Simulation. Pearson Education,					

	ISBN 978-0-138-15037-2 P. Bratley, B. L. Fox, L. E. Schrage: A Guide to Simulation. Springer, ISBN 978-0-387-96467-6 T. T. Allen: Introduction to Discrete Event Simulation and Agent-based Modeling: Voting Systems, Health Care, Military, and Manufacturing. Springer, ISBN 978-0-857-29138-7 A. M. Law: Simulation Modeling & Analysis. McGraw-Hill Professional, ISBN 978-0-071-25519-6

2.11 E-Learning (M-IN-WP-03)

<table>
<tr><td colspan="7"><b>E-Learning / E-Learning (ABR)</b></td></tr>
<tr><td><b>ID</b></td><td><b>Workload</b>
180h</td><td><b>ECTS</b>
6</td><td colspan="2"><b>Term at study start</b>
ST (start): 1
WT (start): 2</td><td><b>Frequency</b>
summer term</td><td><b>Duration</b>
1 term</td></tr>
<tr><td>1</td><td colspan="2"><b>Course</b>
Lecture plus workshops</td><td><b>Contact time lecture</b>
60h</td><td><b>Contact time other</b>
0h</td><td><b>Self-studies</b>
120h</td><td><b>Planned group size</b>
25 students</td></tr>
<tr><td>2</td><td colspan="6"><b>Learning Outcomes</b>
Knowledge of the various users and roles of an LM system and their requirements of the LM system. Ability to analyze the requirements and ability to map the requirements to different services and interfaces. Understanding of the interaction of several user groups and roles in an LM system. Integration of services and basic functionalities into role-specific usage scenarios and corresponding usage interfaces. Assessing an LM system from different perspectives: on the one hand, the user perspective (e.g. as a course author who creates a course fragment) and on the other hand, as a system developer who functionally extends the LM system.</td></tr>
<tr><td>3</td><td colspan="6"><b>Content</b>
The tasks and interaction of the various users and roles of a learning management system (LM system) are presented. The roles of the learners, lecturers, tutors, authors and administrators are elaborated. Their different tasks are considered (e.g. course material management, user, rights and cost management, integration of external resources, etc.). The resulting requirements for an LM system are derived.
Services and interfaces of LM systems are considered. Furthermore, the characteristics of different forms of learning as well as norms and standards in the field of LM systems (SCORM, Dublin Core, LMO, ...) are presented. The learning material lifecycle is taught. The theoretical knowledge is deepened/implemented in two small team phases.
On the one hand, the prototypical creation and integration of an e-learning course fragment into an LM system is carried out. This involves planning and creating course materials. These are modularized, provided with metadata and integrated into an LM system.
The development of LM systems is also considered. For this purpose, either a new functionality to be implemented is identified based on a requirements analysis of a specific user group and then integrated into an LMS, or comparative analyses of existing LMSs are carried out.</td></tr>
<tr><td>4</td><td colspan="6"><b>Course form</b>
4 SWS Seminar-based teaching, practical work on the computer</td></tr>
<tr><td>5</td><td colspan="6"><b>Prerequisites for attending</b>
Formal: none
Content: Multimedia foundations</td></tr>
<tr><td>6</td><td colspan="6"><b>Form of examination</b>
Project incl. Documentation</td></tr>
<tr><td>7</td><td colspan="6"><b>Prerequisites for granting ECTS</b>
Passed exam</td></tr>
<tr><td>8</td><td colspan="6"><b>Utilization of the module (in other studies)</b>
This module is not used in other courses</td></tr>
<tr><td>9</td><td colspan="6"><b>Weight for the final score</b>
Weighting according to the ECTS points</td></tr>
<tr><td>10</td><td colspan="6"><b>Module commissioner:</b> Prof. Dr.-Ing. Mengel
<b>Tutor:</b> Prof. Dr.-Ing. Mengel</td></tr>
</table>

12	**Literature:** Lecture notes for the lecture. - A. Schreiber: CBT-Anwendungen professionell entwickeln, Springer Verlag Wien: Studien Verlag. - R. S. Schifman, G. Heinrich: Multimedia Projektmanagement, Springer Verlag - R. Schulmeister: Lernplattformen für das virtuelle Lernen. Evaluation und Didak-tik. ISBN: 3486272500. R. Oldenbourg Verlag: München u.a. P. Baumgartner et. al.: E-Learning Praxishandbuch: Auswahl von Lernplattformen. Marktübersicht - Funktionen - Fachbegriffe. Innsbruck-Wien: Studien Verlag

2.12 Safe and Secure Programming in Rust (M-IN-WP40)

Safe and Secure Programming in Rust, Sicheres und geschütztes Programmieren in Rust (RUST)						
ID	**Workload** 180h	**ECTS** 6	**Term at study start** ST (start): 2 WT (start): 1		**Frequency** winter term	**Duration** 1 term
1	**Course** Lecture and Tutorials, practical project		**Contact time lecture** 30h	**Contact time other** 30h	**Self-studies** 120h	**Planned group size** 25 students
2	**Learning Outcomes** By successful completion of this course, students obtain the following skills: - They internalized that programming in safety-critical domains is fundamentally different from programming in 'regular' domains. - They understand how strict programming languages can contribute to safe and secure programming. - They can apply basic and advanced concepts of the Rust programming language in practical projects. - They can build robust Rust applications for use in safety-critical domains.					
3	**Content** Malfunctions of software in safety-critical systems as well as cyberstrikes can lead to severe losses including death and environmental harm. Hence, when building software for such environments the use of safe and secure programming languages is essential. One suitable programming language for this use case is Rust. Moreover, Rust is also continuously gaining popularity and is used in leading open source projects such as the Linux kernel or the Firefox browser. Rust is particularly attractive because it enables both system-level and application-oriented programming while pursuing the goal of making programs safe and secure. The first part of this course will start with an introduction to safety-critical systems. Afterwards, the basics of Rust (syntax, concepts) will be explained and comparisons to other programming languages (e.g. Java or C/C++) will be drawn. Here, the focus will be on memory management without a garbage collector and its implications on safety and security. In the second part of this course, the participants will deepen the theory through practical work on real development projects. The course follows the concept of 'research-based learning' and therefore requires an adequate degree of initiative and willingness to learn. In particular, we expect that students learn independently by means of designated tutorials.					
4	**Course form** 2 SWS Lecture, 2 SWS Tutorial					
5	**Prerequisites for attending** Formal: none Content: none					
6	**Form of examination** Written exam, Oral exam, Project Work					
7	**Prerequisites for granting ECTS** Passed exam					
8	**Utilization of the module (in other studies)** This module is not used in other courses					
9	**Weight for the final score** Weighting according to the ECTS points					
10	**Module commissioner:** Prof. Dr. Daniel Kulesz **Tutor**: Prof. Dr. Daniel Kulesz					

12	**Literature:** - “Programming Rust: Fast, Safe Systems Development”, Jim Blandy, Jason Orendorff, Leonora Tindall, 2nd. ed, 2021, O'Reily - “Embedded software development for safety-critical systems”, Chris Hobbs, 2nd ed., 2020, CRC Press

3 Complementary Modules (Comprehensive)

3.1 Advanced Project Management (M-IN-WP01)

Advanced Project Management, Fortgeschrittenes Projektmanagement (PROJM)						
ID	**Workload** 180h	**ECTS** 6	**Term at study start** ST (start): 2 WT (start): 1		**Frequency** winter term	**Duration** 1 term
1	**Course** Seminar		**Contact time lecture** 30h	**Contact time other** 30h	**Self-studies** 120h	**Planned group size** 25 students
2	**Learning Outcomes** Students acquire skills for planning and managing complex projects from science, industry and society. They are familiar with the essential process models and methods, know their specific characteristics and areas of application. They develop the ability to independently plan, organize and manage software development projects. Students will be able to prepare feasibility studies, resource estimates and effort estimates and draw conclusions from them. They will be able to analyze and evaluate risks and safety-related areas for projects. Students develop teamwork skills and the ability to solve problems independently. Students master the mechanisms of agile project execution and are able to implement and apply them.					
3	**Content** Students acquire skills for planning and managing complex projects from science, industry and society. They are familiar with the essential process models and methods, know their specific characteristics and areas of application. They develop the ability to independently plan, organize and manage software development projects. Students will be able to prepare feasibility studies, resource estimates and effort estimates and draw conclusions from them. They will be able to analyze and evaluate risks and safety-related areas for projects. Complexity considerations of large software systems - Process models of software development (V-model, RUP, Extreme Programming, Scrum etc.) - Application of process models and their specific characteristics, - Planning techniques and checklists for project planning - Tools and aids for project management - Tracking of requirements from analysis to implementation - Change and configuration management - Time management and resource management - Project management standards - Effort estimation (function point analysis and others) - Metrics based process management and control.					
4	**Course form** seminaristic					
5	**Prerequisites for attending** Formal: none Content: none					
6	**Form of examination** Oral examination, presentation					
7	**Prerequisites for granting ECTS** Passed exam					
8	**Utilization of the module (in other studies)** This module is not used in other courses					
9	**Weight for the final score** Weighting according to the ECTS points					
10	**Module commissioner:** Prof. Dr. Thomas Marx **Tutor:** Prof. Dr. Thomas Marx					

12	**Literature:** Höhn, Reinhard; Höppner, Stephan, Das V-Modell XT, Grundlagen, Methodik und Anwendungen, Springer, jeweils aktuelle Ausgabe Wolf, Henning, Roock, Stefan, Lippert, Martin, eXtreme Programming: Eine Einführung mit Empfehlungen und Erfahrungen aus der Praxis, Dpunkt, jeweils aktuelle Ausgabe Pichler, Roman, Scrum - Agiles Projektmanagement erfolgreich einsetzen, Dpunkt. jeweils aktuelle Ausgabe, ISBN10 3898644782 Verstegen, Gerhard. Projektmanagement mit dem Rational Unified Process. Springer. Berlin. 2008. Ebel, Nadin. PRINCE2:2009 - für Projektmanagement mit Methode. Addison-Wesley. München. jeweils aktuelle Ausgabe. A Guide to the Project Management Body of Knowledge. Project Management Institute. jeweils aktuelle Ausgabe. Function Point Analyse Poensgen, Benjamin; Bock, Bertram. Die Function-Point-Analyse: Ein Praxishandbuch. dpunkt Verlag. 2005. Hindel, Bernd; Hörmann, Klaus; Müller, Markus; Schmied, Jürgen. Basiswissen Software-Projektmanagement. dpunkt.verlag. jeweils aktuelle Ausgabe

3.2 Innovation and IT (M-IN-WP39)

<table>
<tr><td colspan="6"><b>Innovation & IT, Innovation und IT (INOV)</b></td></tr>
<tr><td><b>Workload</b>
180h</td><td><b>ECTS</b>
6</td><td colspan="2"><b>Term at study start</b>
ST (start): 1
WT (start): 2</td><td><b>Frequency</b>
summer term</td><td><b>Duration</b>
1 term</td></tr>
<tr><td colspan="2"><b>Course</b>
Seminar</td><td><b>Contact time lecture</b>
30h</td><td><b>Contact time other</b>
30h</td><td><b>Self-studies</b>
120h</td><td><b>Planned group size</b>
25 students</td></tr>
<tr><td colspan="6"><b>Learning Outcomes</b>
Students know and recognize basic digital economy concepts and IS-based business models. They are familiar with ideas concerning the application of IS-based innovations, networks and platforms for communication, inter- action and transaction in a globalized world and can analyze and apply them.
Students are aware of the digital economy’s main innovative concepts, methods, and instruments. Students are able to distinguish IS-based business model applications, implementations, and innovations. They are able to reflect, analyze, discuss and apply those concepts. Students are able to assess the value of digital business, trans- formation, and the economics of digitization. They are capable of assessing applied practical implementations in a competent way. Students recognize business transformations induced by IS innovations, and are able to reflect and apply concepts and models to actual cases by design. They are capable of reflecting potential social and cultural impacts and gain knowledge in a self-directed manner.

Due to a comprehensive statement of current topics students gain broad knowledge. In-depth insights into innovative best demonstrated available technology (such as big data and business analysis) and its business application deepen their knowledge. Decision-making under uncertain conditions is required.
Students team up in small groups and are able to lead small teams in a responsible way, research and apply knowledge in a self-directed manner, and discuss their results. They are able to promote professional development of their fellow students’ appropriate knowledge and discuss their results with peers and with experts.
Self-motivation/self-study
• Homework/Exercise (Breadth)
Fundamental concepts of economic decisions (eg value chains and business systems) are repeated. Concepts of innovation management have to be read, analyzed and discussed.
• Homework / Exercise (Depth)
Important topics such as neo-mediation or disintermediation are prepared in self-study and subsequently discussed in detail.</td></tr>
<tr><td colspan="6"><b>Content</b>
Innovation, digital economy, transformation classification in a scientific context
Current topics and best demonstrated available IS-technology
Terminology, concepts and models: innovation, digital economy, transformation, and IS-based business models
Selected case studies
Applied digital economy, transformation applications
Trends (e. g. mobile business)
Social and cultural context and impact</td></tr>
<tr><td colspan="6"><b>Course form</b>
Seminar</td></tr>
<tr><td colspan="6"><b>Prerequisites for attending</b>
Formal: none
Content: none</td></tr>
<tr><td colspan="6"><b>Form of examination</b>
Written examination in the form of a self-directed project including presentation (100 %)</td></tr>
<tr><td colspan="6"><b>Prerequisites for granting ECTS</b>
Passed Exam</td></tr>
</table>

Utilization of the module (in other studies)
This module is not used in other courses

Weight for the final score
Weighting according to the ECTS points

Module commissioner: Prof. Dr. Bernhard Ostheimer
Tutor: Prof. Dr. Bernhard Ostheimer

Literature:
Christensen, C. M.: The Innovator’s Dilemma. Boston, MA, USA, Harvard Business Review Press
Clement, R., Schreiber, D.: Internet-Ökonomie – Grundlagen und Fallbeispiel der vernetzten Wirtschaft. Berlin, Springer Gabler
Day, G. S.; Moorman, C.: Strategy from the Outside in. London, McGraw-Hill
Kaufmann, T.: Geschäftsmodelle in Industrie 4.0 und dem Internet der Dinge. Berlin, Springer Vieweg
Kollmann, T.: E-Business. Berlin, Springer Gabler
Laudon, K. C.; Traver, C. G.: E-Commerce 2016: Business, Technology, Society. Upper Saddle River, NJ, USA, Pearson
Osterwalder, A.; Pigneur, Y.: Business Model Generation. Hoboken, NJ, USA, John Wiley & Sons
Rogers, D. L.: Digital Transformation Playbook: Rethink Your Business for the Digital Age. New York, Columbia University Press
Westerman, G.; Bonnet, D.; McAfee, A.: Leading Digital: Turning Technology into Business Transformation. Boston, MA, USA, Harvard Business Review Press
Wirtz, B. W.: Electronic Business. Berlin, Springer Gabler
Most recent edition.
Relevant journal articles, e.g.:
Gimpel, H.; Röglinger, M. (2015): Digital Transformation: Changes and Chances – Insights based on an Empirical Study. Fraunhofer Institute for Applied Information Technology
Hansen, R.; Sia, S. K. (2015): Hummel's Digital Transformation Toward Omnichannel Retailing: Key Lessons Learned. MIS Quarterly Executive, Vol. 14, Issue 2
Kane, G. C.; Plamer, D.; Phillips, A. N.; Kiron, D.; Buckley, N. (2015): Strategy, not Technology, Drives Digital Transformation. MIT Sloan Management Review and Deloitte University Press
Matt, C.; Hess, T.; Benlian, A. (2015): Digital Transformation Strategies; Business & Information Systems Engineering, Vol. 57, Issue 5

3.3 Advanced Software Engineering: Principles & Structures (M-IN-WP38)

<table>
<tr><td colspan="7"><b>Advanced Software Engineering: Principles & Structures, Fortgeschrittenes Software-Engineering: Prinzipien und Strukturen (ADSE)</b></td></tr>
<tr><td><b>ID</b></td><td><b>Workload</b>
180h</td><td><b>ECTS</b>
6</td><td colspan="2"><b>Term at study start</b>
ST (start): 2
WT (start): 1</td><td><b>Frequency</b>
winter term</td><td><b>Duration</b>
1 term</td></tr>
<tr><td>1</td><td colspan="2"><b>Course</b>
Seminar</td><td><b>Contact time lecture</b>
30h</td><td><b>Contact time other</b>
30h</td><td><b>Self-studies</b>
120h</td><td><b>Planned group size</b>
25 students</td></tr>
<tr><td>2</td><td colspan="6"><b>Learning Outcomes</b>
The students know advanced topics and interrelationships in the subject areas of software engineering: Requirements engineering, specification as well as system architecture, development processes and related aspects of quality and security. The students therefore are familiar with modern principles and paradigms in the field of software design, development, deployment and operation. They can apply this knowledge to practical problems. The analysis, design and development of software systems can be actively accompanied by the students as part of a leading team. They also know how to coordinate the activities in the development and deployment chain of large software systems and are able to assess technical and economic risks as well as software quality.

Competencies
The course covers aspects of the development process from the determination of requirements to quality assurance. Students use common platforms, frameworks and tools to train their ability to plan, monitor and control large complex projects.
Working on questions in small groups trains in dealing with conflicting goals, promotes discussion, critical faculties and presentation.
The module contributes in particular to the development of leadership competences. The handling of case studies and case studies promotes the necessary decision-making competence. In addition, rhetorical skills and the ability to convince and motivate employees are of great importance.

Self-motivation/self-study
- Homework / Exercise (Width)
After an introductory presentation, the students work independently into concrete projects and gain in particular an impression of the complexity.
- Homework / Exercise (Design)
The students develop specific, corresponding solutions for selected questions in software engineering and develop concrete implementation approaches.
All previously during the study program acquired knowledge is brought together here</td></tr>
<tr><td>3</td><td colspan="6"><b>Content</b>
Requirements engineering: methods and processes for the definition, documentation and management of functional and non-functional requirements.
- Software architecture: design and construction principles, paradigms and structural styles (like microservices), reference architectures, frameworks and libraries.
- Software development process: management of complex software development projects, management of soft- ware product lines, versioning, prototyping, agile methods
- Software deployment, delivery and operating: Common and crucial aspects of the deployment, delivery and operating chain of software systems as far as these are associated to software engineering: e.g. container, distributed systems, cloud computing, software as a service (SaaS), edge and fog computing</td></tr>
<tr><td>4</td><td colspan="6"><b>Course form</b>
Seminaristic</td></tr>
</table>

5	**Prerequisites for attending** Formal: none Content: none
6	**Form of examination** Written examination in the form of a self-directed project including presentation (presentation 40% /documentation 60%)
7	**Prerequisites for granting ECTS** Passed exam
8	**Utilization of the module (in other studies)** This module is not used in other courses
9	**Weight for the final score** Weighting according to the ECTS points
10	**Module commissioner:** Prof. Dr. Jens Reinhardt **Tutor:** Prof. Dr. Jens Reinhardt
12	**Literature:** Sommerville, I. Software Engineering, Pearson. Most recent edition.

3.4 Business Models and IT-Strategy (M-IN-WP37)

Business Models and IT-Strategy, Geschäftsmodelle und IT-Strategie (BMST)						
ID	**Workload** 180h	**ECTS** 6	**Term at study start** ST (start): 2 WT (start): 1		**Frequency** winter term	**Duration** 1 term
1	**Course** Lecture		**Contact time lecture** 30h	**Contact time other** 30h	**Self-studies** 120h	**Planned group size** 25 students
2	**Learning Outcomes** By developing different business models and identifying appropriate IT strategies, the students assess critically possible scenarios by means of discussions, current case studies and research approaches. The students know how companies can take advantage of changes in the market through appropriate transformations of value chains and business systems to their advantage. They can develop IT strategies that support the company's objectives or enable specific business models. Objectives and architectures of inter-company networking can be explained using current examples from various sectors. In particular, they understand the role that IT can play as a differentiating factor in the implementation of innovative business models, and the impact of IT innovations on the business and IT strategy. Competencies The module contributes in particular to the development of leadership competences. The handling of case studies and case studies promotes the necessary decision-making competence. In addition, rhetorical skills and the ability to convince and motivate employees are of great importance. Self-motivation/self-study • Homework / Exercise (width) After an introductory presentation, the students work independently into concrete business models as well as corresponding IT strategies and gain in particular an impression of the complexity. • Homework / Exercise (Design) The students develop specific, corresponding IT strategies for selected business models and develop concrete implementation approaches. All previously during the study program acquired knowledge is brought together here					
3	**Content** Development of IT strategy and alignment with the business strategy Business models and development strategies Relationship between business model and IT strategy Case studies Typical examples of content are: • ICT Governance: Targeting the IT strategy according to the business strategy, e.g. by means of Control Objecti- ves for Information and Related Technology (COBIT). • B2B and B2C scenarios, e.g. Integrated Producrement, Collaboration Networks, Mass Customization. • Case studies: IT as an enabler of innovative business models					
4	**Course form** Seminar (lecture, practical parts, self-learning/study hours)					
5	**Prerequisites for attending** Formal: none Content: none					

6	**Form of examination** Written examination in the form of a self-directed project (business model and corresponding IT strategy) including presentation (Presentation 40% /documentation 60%)
7	**Prerequisites for granting ECTS** Passed exam
8	**Utilization of the module (in other studies)** This module is not used in other courses
9	**Weight for the final score** Weighting according to the ECTS points
10	**Module commissioner:** Prof. Dr. Anett Mehler-Bicher **Tutor:** Prof. Dr. Anett Mehler-Bicher
12	**Literature:** Becker, J.; Knackstedt, R.; Pfeiffer, D.: Wertschöpfungsnetzwerke, Physica. Buchta, D.; Eul, M.; Schulte-Croonenberg, H.: Strategisches IT Management, Gabler. Gassmann, O.; Frankenberger, K; Csik, M.: Geschäftsmodelle entwickeln, Hanser Osterwalder, A.; Pigneur, Y. Business Model Generation Keuper, F.; Schomann, M.; Grimm, R.: Strategisches IT Management. Management von IT und IT gestütztes Management, Gabler. McKeen, J.D.; Smith, H.: IT Strategy. Prentice Hall. Most recent edition.

3.5 Optimization and Operations Research (M-IN-WP38)

Optimization and Operations Research, Optimierung und Operations Research (OPOR)						
ID	**Workload** 180h	**ECTS** 6	**Term at study start** ST (start): 1 WT (start): 2		**Frequency** summer term	**Duration** 1 term
1	**Course** Lecture		**Contact time lecture** 30h	**Contact time other** 30h	**Self-studies** 120h	**Planned group size** 25 students
2	**Learning Outcomes** Students know different modeling methods for the presented set of problem classes. They understand the solution algorithms for these models and know their advantages and limitations. Students can model new problems and implement solution methods for these models.					
3	**Content** - Linear optimization - Simplex algorithm - Integer Optimization - Branch & Bound, Branch & Cut - Network Problems - Scheduling Problems - Heuristics & Meta Heuristics					
4	**Course form** 2 SWS Lecture, 2 SWS Turorial					
5	**Prerequisites for attending** Formal: none Content: Bachelor’s Mathematics 1 and 2 plus Algorithms and Data Structure					
6	**Form of examination** Written exam Oral exam Presentation Term paper					
7	**Prerequisites for granting ECTS** Passed exam					
8	**Utilization of the module (in other studies)** This module is not used in other courses					
9	**Weight for the final score** Weighting according to the ECTS points					
10	**Module commissioner:** Prof. Dr. Florian Dahms **Tutor**: Prof. Dr. Florian Dahms					
12	**Literature:** Alexander Schrijver: Theory of Linear and Integer Programming					

3.6 Model-Based Software Engineering (M-IN-WP39)

Model-Based Software Engineering, Modell basiertes Software Engineering (MBSE)						
ID	**Workload** 180h	**ECTS** 6	**Term at study start** ST (start): 2 WT (start): 1		**Frequency** winter term	**Duration** 1 term
1	**Course** Lecture and Tutorials, practical project		**Contact time lecture** 30h	**Contact time other** 30h	**Self-studies** 120h	**Planned group size** 25 students
2	**Learning Outcomes** By the end of this module, students will be able to: - Identify and describe the fundamental concepts and terminology of model-based software engineering. - Explain the significance of modeling in software engineering. - Create requirements models (context models, goal models, scenarios) - Create static and dynamic architecture models - Evaluate different modeling approaches and tools and choose the appropriate one for a given software project. - Critically assess the quality of software models - Develop comprehensive models for complex software systems					
3	**Content** • Introduction to Model-Based Software Engineering o Overview of model-based software engineering o Benefits and challenges of model-based software engineering o Key concepts and terminology • Modeling Requirements o Context modeling o Goal modeling o Scenario modeling • Modeling Software Architecture o Static architecture models o Dynamic architecture models • Analyzing and Modeling Variant-Intensive Systems o Identifying variability in software system o Approaches to modeling variant-intensive systems • Model-Based Quality Assurance o Validation and verification approaches for model-based artifacts o Model-based testing					
4	**Course form** 2 SWS Lecture, 2 SWS Tutorial and Practical Project					
5	**Prerequisites for attending** Formal: none Content: Basic software engineering knowledge					
6	**Form of examination** Written exam, Oral exam, Project Work					
7	**Prerequisites for granting ECTS** Passed exam					
8	**Utilization of the module (in other studies)** This module is not used in other courses					
9	**Weight for the final score** Weighting according to the ECTS points					

10	**Module commissioner:** Prof. Dr. Jennifer Brings **Tutor**: Prof. Dr. Jennifer Brings
12	**Literature:** - Brambilla, M., Cabot, J., Wimmer, M. (2022). Model-Driven Software Engineering in Practice. Germany: Springer International Publishing. - Rumpe, B. (2017). Agile Modeling with UML: Code Generation, Testing, Refactoring. Germany: Springer International Publishing. - Burgueño, L., Ciccozzi, F., Famelis, M. et al. Contents for a Model-Based Software Engineering Body of Knowledge. Softw Syst Model 18, 3193–3205 (2019). https://doi.org/10.1007/s10270-019-00746-9 - Kautz, O., Roth, A., Rumpe, B. (2018). Achievements, Failures, and the Future of Model-Based Software Engineering. In: Gruhn, V., Striemer, R. (eds) The Essence of Software Engineering. Springer, Cham. https://doi.org/10.1007/978-3-319-73897-0_13 - Selic, B. (2008) Personal reflections on automation, programming culture, and model-based software engineering. Autom Softw Eng 15, 379–391. https://doi.org/10.1007/s10515-008-0035-7 - Ciccozzi, F., et al. (2018). Towards a body of knowledge for model-based software engineering. In Proceedings of the 21st ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings (MODELS '18). Association for Computing Machinery, New York, NY, USA, 82–89. https://doi.org/10.1145/3270112.3270121 - Broy, M. (2011). Seamless Method- and Model-based Software and Systems Engineering. In: Nanz, S. (eds) The Future of Software Engineering. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-15187-3_2 - Pohl, K., Böckle, G. and Linden, F. v. d. (2005) Software Product Line Engineering - Foundations, Principles, and Techniques, Springer. - ITU (2018). Recommendation Z.151: User Requirements Notation (URN) - Language Definition, Technical report, International Telecommunication Union. - ITU (2011) Recommendation Z.120: Message Sequence Chart (MSC), Technical report, International Telecommunication Union. - OMG (2017) Unified Modeling Language. Technical report, Object Management Group.