Technology
Arts Sciences
TH Köln

Course Manual EKS

Development of Complex Software Systems

Version: 1 | Last Change: 03.09.2019 11:28 | Draft: 0 | Status: vom verantwortlichen Dozent freigegeben

General information

Long name	Development of Complex Software Systems
Approving CModule	EKS_BaTIN
Responsible	Prof. Dr. Hans Nissen Professor Fakultät IME
Valid from	winter semester 2022/23
Level	Bachelor
Semester in the year	winter semester
Duration	Semester
Hours in self-study	60
ECTS	5
Professors	Prof. Dr. Hans Nissen Professor Fakultät IME
Requirements	Specification and modeling of systems and software with UML, modularization in Java, simple design patterns, basic software testing methode, various architectures of systems and software, basic quality assurance concepts, version management skills, very good practical and theoretical knowledge of the programming language Java

Literature

E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns, MITP Verlags GmbH & Co. KG, 2015.

R. C. Martin: Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall, 2008.

S. McConnell: Code Complete, Microsoft Press, 2. Auflage, 2004.

M. Fowler: Refactoring: Improving the Design of Existing Code. Addison-Wesley Verlag, 2. Auflage, 2018.

G. Oelmann: Modularisierung mit Java 9, dpunkt Verlag, 2018.

R.S. Hull, K. Pauls, S. McCulloch, D. Savage: OSGi in Action, Manning Publications, 2011.

G. Wütherich, N. Hartmann, B. Kolb, M. Lübken: Die OSGi Service Plattform, dpunkt Verlag, 2008.

A. Spillner, T. Linz: Basiswissen Softwaretest, dpunkt Verlag, 5. Auflage, 2012

P. Liggesmeyer: Software-Qualität: Testen, Analysieren und Verifizieren von Software, Spektrum Akademischer Verlag, 2. Auflage, 2009.

H.M. Sneed, M. Winter: Testen objektorientierter Software, Hanser Verlag, 2001.

Final exam

Language	German
Separate final exam	Yes

Details	oral exam, with many students written exam The oral exam or written exam ensures that each student also individually fulfills the goals of the Learning Outcome has reached, through tasks of the following types: questions about basic knowledge of design principles, architectural concepts, test procedures, application of design patterns to given problem cases, design or extension of a modularized system architecture with supporing specified non-functional properties, creation of suitable logical test specifications and concrete test cases.
Minimum standard	At least 50% of the total number of points.
Exam Type	EN mündliche Prüfung, strukturierte Befragung

− Lecture / Exercises

Learning goals

Goal type	Description
Knowledge	design patterns
Knowledge	modularization
Knowledge	professional code development
Knowledge	advanced Java concepts
Knowledge	Module-oriented architectural principles
Knowledge	complex test procedures
Skills	apply and evaluate design patterns
Skills	Apply and evaluate approaches to professional code development
Skills	Apply automated code analysis methods and interpret the results
Skills	Design and implement modularized architectures
Skills	use complex test procedures

Special requirements

keine

Accompanying material	electronic presentation slides for the lecture, electronic worksheets for exercises
Separate exam	No

Expenditure classroom teaching

Type	Attendance (h/Wk.)
Lecture	2
Exercises (whole course)	1
Exercises (shared course)	1
Tutorial (voluntary)	0

− Practical training

Learning goals

Goal type	Description
Skills	implemtation of design pattern
Skills	Create modularized architectures for large-scale applications
Skills	apply automated code review and static code analysis
Skills	Select test method and apply to programs

Expenditure classroom teaching

Type	Attendance (h/Wk.)
Practical training	1
Tutorial (voluntary)	0

Special requirements

keine

Accompanying material	electronic presentation slides for the lecture, electronic worksheets for exercises
Separate exam	Yes

Separate exam

Exam Type	EN praxisnahes Szenario bearbeiten (z.B. im Praktikum)

Details	Students work in small teams. Each group completes several internship sessions with assigned laboratory appointments. In each session, programming tasks are solved. To prepare for a lab appointment, a homework sheet has to be solved. The developed solutions must be submitted by the students before the lab date and explained and defended during the appointment to the supervisor (K.16). If this test is not passed, then a repetition task must be edited and presented until a follow-up appointment; in the case of recurrence, this leads to failure of the internship. In addition, during the laboratory appointment, an attendance sheet with further tasks has to be solved under supervision (and possibly with assistance) in a controlled environment.
Minimum standard	Successful participation in all laboratory appointments, i.e. in particular independent solution (or with some assistance if necessary) of the assignments.