

Modulhandbuch EKS

Entwicklung komplexer Software-Systeme

Bachelor Technische Informatik 2020

Version: 1 | Letzte Änderung: 03.09.2019 11:27 | Entwurf: 0 | Status: vom Modulverantwortlichen freigegeben |
Verantwortlich: Nissen

– Allgemeine Informationen

Anerkannte Lehrveranstaltungen	EKS Nissen
Gültig ab	Wintersemester 2022/23
Modul ist Bestandteil der Studienschwerpunkte	NVS - Netze und Verteilte Systeme SOS - Software-Systeme
Dauer	1 Semester
ECTS	5
Zeugnistext (de)	Entwicklung komplexer Software-Systeme
Zeugnistext (en)	Development of Complex Software Systems
Unterrichtssprache	deutsch
abschließende Modulprüfung	Ja

Modulprüfung

Benotet	Ja
Konzept	mündliche Prüfung, bei vielen Studenten schriftliche Klausur Die mündliche Prüfung bzw. schriftliche Klausur stellt sicher, dass jeder Studierende auch individuell die Ziele des Learning Outcomes erreicht hat, durch Aufgaben der folgenden Typen: Fragen zu Grundwissen über Entwurfsprinzipien, Architekturkonzepten, Testverfahren (K.3, K.9), Anwendung von Entwurfsmustern auf gegebene Problemfälle (K.5, K.9), Entwurf oder Erweiterung einer modularisierten Systemarchitektur mit Gewährleistung vorgegebenen nicht-funktionaler Eigenschaften (K.1, K.3, K.4, K.5) Erstellung geeigneter logischer Testspezifikationen und konkreter Testfälle (K.2, K.7, K.9)
Frequenz	Jedes Semester

– Allgemeine Informationen

Inhaltliche Voraussetzungen

SE - Software Engineering Spezifikation und Modellierung von Systemen und Software mit UML, Modularisierung in Java, einfache Entwurfsmuster, grundlegende Verfahren zum Prüfen von Software, verschiedene Architekturen von Systemen und Software, Grundbegriffe der Qualitätssicherung, Kenntnisse in Versionsverwaltung

PI1 - Praktische Informatik 1 sehr gute praktische und theoretische Kenntnisse der Programmiersprache Java

PI2 - Praktische Informatik 2 sehr gute praktische und theoretische Kenntnisse der Programmiersprache Java

PP - Programmierpraktikum sehr gute praktische und theoretische Kenntnisse der Programmiersprache Java

Handlungsfelder

Systeme zur Verarbeitung, Übertragung und Speicherung von Informationen für technische Anwendungen planen, realisieren und integrieren

Anforderungen, Konzepte und Systeme analysieren und bewerten

Informationstechnische Systeme und Prozesse organisieren und betreiben

Learning Outcomes

ID	Learning Outcome
LO1	<p>Was: Das Modul vermittelt unterschiedliche Kompetenzen, die für die Entwicklung und Pflege komplexer Software-Systeme erforderlich sind: die Anwendung und Beurteilung von Entwurfsmustern (K.1, K.3 K.4, K.5, K.9, K.10), die Anwendung von Ansätzen zur professionellen Code-Entwicklung (K.6, K.9), der Einsatz und die Beurteilung von Verfahren zur statischen Code-Analyse (K.4, K.7, K.9), die Beherrschung fortgeschrittene Java-Konzepte (K.5, K.6), der Entwurf und die Realisierung modularisierter Software-Architekturen (K.1, K.3, K.5, K.10), die Einordnung und der Einsatz komplexer Testverfahren (K.2, K.7, K.9).</p> <p>Womit: Der Dozent vermittelt Wissen und Basisfertigkeiten in einem Vorlesungs/Übungsteil unter Verwendung einer Fallstudie und verschiedenen praktischen Demonstrationen. Im zugehörigen Praktikum erarbeiten die Studierenden in Kleingruppen Lösungen zu vorgegebenen Problemen und wenden dabei die Themenbereiche der Vorlesung praktisch an. Hierzu wird eine selbständige Vertiefung einzelner Themenbereiche, insbesondere die Verwendung typischer Werkzeuge, verlangt (K.8, K.9, K.15).</p> <p>An den Präsenzterminen müssen die Studierenden ihre Lösungen erläutern und verteidigen (K.16).</p> <p>Wozu: Bei der Entwicklung und Pflege moderner Software muss man sich mit einer stetig zunehmenden</p>

System-Komplexität auseinandersetzen. Einige typische Gründe für eine hohe Komplexität sind: die Systeme sind sehr umfangreich, es bestehen sehr viele Schnittstellen zu anderen Systemen, es werden viele und zum Teil sehr umfangreiche Frameworks eingesetzt. Für eine erfolgreiche Tätigkeit in diesem Umfeld ist die sichere Beherrschung hierauf ausgerichteter Methoden, Konzepte und Technologien unbedingt erforderlich (HF.1). Die Erweiterung eines komplexen Systems erfordert die umfangreiche Analyse der bestehenden Abhängigkeiten und die Beurteilung der unterschiedlichen Entwurfalternativen bezüglich ihrer Auswirkungen auf das Gesamtsystem (HF.2). Ein wesentliches Hilfsmittel zur Beherrschung der System-Komplexität stellt die Wahl einer geeigneten Organisationsform der Systems dar (HF.3). Hierfür sind umfangreiche theoretische und praktische Kenntnisse von Organisationsformen im Kleinen (z.B. Entwurfsmuster) und im Großen (z.B. Modularisierung) erforderlich.

Kompetenzen

Kompetenz	Ausprägung
Typische Werkzeuge, Standards und Best Practices der industriellen Praxis kennen und einsetzen	diese Kompetenz wird vermittelt
In Systemen denken	diese Kompetenz wird vermittelt
Konzepte und Methoden der Informatik, Mathematik und Technik kennen und anwenden	diese Kompetenz wird vermittelt

fachliche Probleme abstrahieren und formalisieren	diese Kompetenz wird vermittelt
---	---------------------------------

Systeme entwerfen	diese Kompetenz wird vermittelt
-------------------	---------------------------------

Systeme realisieren	diese Kompetenz wird vermittelt
---------------------	---------------------------------

In vorhandene Systeme einarbeiten und vorhandene Komponenten sinnvoll nutzen	diese Kompetenz wird vermittelt
--	---------------------------------

Systeme analysieren	diese Kompetenz wird vermittelt
---------------------	---------------------------------

Systeme prüfen	diese Kompetenz wird vermittelt
----------------	---------------------------------

Informationen beschaffen und auswerten; Technische Zusammenhänge darstellen und erläutern	diese Kompetenz wird vermittelt
---	---------------------------------

Befähigung zum lebenslangen Lernen	diese Kompetenz wird vermittelt
------------------------------------	---------------------------------

Kommunikative und interkulturelle Fähigkeiten anwenden	diese Kompetenz wird vermittelt
--	---------------------------------

– Vorlesung / Übungen

Typ	Vorlesung / Übungen
------------	---------------------

Separate Prüfung	Nein
-------------------------	------

Exemplarische inhaltliche Operationalisierung	Anhand praxisnaher Beispiele und einer Fallstudie werden typische Entwurfsmuster entwickelt und diskutiert. Zur Verbesserung der individuellen Programmierfähigkeiten werden Ansätze zur professionellen Code-Entwicklung vorgestellt und kritisch bewertet. Zusätzlich werden unterschiedliche Verfahren zur automatisierten Code-Analyse charakterisiert, erläutert und praktisch angewendet. Unterschiedliche Ansätze zur Modularisierung von Systemen werden gegenübergestellt, deren Anwendung in Vorführungen erläutert und in Übungen praktisch angewendet. Fortgeschrittene Testverfahren werden eingeführt und in Übungen angewendet.
--	--

– Praktikum

Typ	Praktikum
------------	-----------

Separate Prüfung	Ja
-------------------------	----

Exemplarische inhaltliche Operationalisierung	Die Anwendung von Entwurfsmustern erfolgt durch die Realisierung spezifizierter Systemteile. Auf Basis einer gegebenen Spezifikation kann die Erstellung entsprechender modularisierter Architekturen erfolgen. Für ein gegebenes Programmbeispiel sollen geeignete Testverfahren ausgewählt und umgesetzt werden. Für die selbst erstellten Programme werden automatisierte Code-Reviews und statische Code-Analysen durchgeführt.
--	---

Separate Prüfung

Benotet	Nein
----------------	------

Frequenz	Einmal im Jahr
-----------------	----------------

Voraussetzung für Teilnahme an Modulprüfung	Ja
--	----

Konzept	Die Studierenden schließen sich zu Kleingruppen zusammen. Jede Kleingruppe absolviert mehrere Praktikumssitzungen mit zugewiesenen Laborterminen. In jeder Sitzung werden Programmieraufgaben gelöst (K.6, K.10). Zur Vorbereitung eines Labortermins muss ein Hausaufgabenblatt praktisch gelöst werden. Die erarbeiteten Lösungen müssen die Studierenden vor dem Labortermin abgeben und am Termin gegenüber dem Betreuer erläutern und verteidigen (K.16). Wird diese Prüfung nicht bestanden, so muss eine Wiederholungsaufgabe bis zu einem Folgetermin bearbeitet und dort präsentiert
----------------	---

werden;
im Wiederholungsfall führt dies
zum Nichtbestehen des
Praktikums.

Zusätzlich wird während des
Labortermins ein
Anwesenheitsblatt mit weiteren
Aufgaben unter
Aufsicht (und ggf. mit
Hilfestellung) in einer
kontrollierten Umgebung
bearbeitet.

Hierdurch stellt jede Kleingruppe
ihre Fähigkeit zur selbständigen
Lösung
unter Beweis.

Im Praktikum werden die
folgenden Typen von Aufgaben
bearbeitet:

Umsetzung von Entwurfsmustern
für ein gegebenes Problem (K.3,
K.5, K.6),

Entwurf und Realisierung einer
modularisierten

Systemarchitektur für ein
gegebenes Anwendungsszenario
(K.1, K.3, K.4, K.5, K.6),

Anwendung fortgeschrittener
Testverfahren auf Programmcode
(K.3, K.7),

Analyse eines Programms unter
Verwendung von automatischen
Code-Review und statischer
Code-Analyse (K.4, K.7) und
Entwurf und Realisierung eines
Programms unter Verwendung
fortgeschrittener Java-Konzepte
(K.5, K.6).