Technology
Arts Sciences
TH Köln

Modul

QEKS - Qualitätsgesteuerter Entwurf komplexer Softwaresysteme

Master Technische Informatik 2020

Version: 3 | Letzte Änderung: 06.10.2019 17:26 | Entwurf: 0 | Status: vom Modulverantwortlichen freigegeben | Verantwortlich: Kreiser

Allgemeine Informationen

Anerkannte Lehrveranstaltungen	SEKM_Kreiser
Dauer	1 Semester
ECTS	5
Zeugnistext (de)	Qualitätsgesteuerter Entwurf komplexer Softwaresysteme
Zeugnistext (en)	Quality Controlled Development of Complex Software Systems
Unterrichtssprache	deutsch oder englisch
abschließende Modulprüfung	Ja

Modulprüfung

Benotet	Ja
Frequenz	Jedes Semester

Prüfungskonzept

Mündliche Prüfung nach schriftlicher Vorbereitung.
Anhand einer realitätsnahen Aufgabenstellung angemessener Komplexität entwickeln und modellieren die Studierenden eine geeignete Softwarearchitektur für ein verteiltes Automatisierungssystem unter angemessener Anwendung von Strategien zur Wiederwendung von Modell- und/oder Softwareartefakten. Sie begründen die essenziellen Strukturen ihrer Architektur unter Bezugnahme auf die spezifische Zielsetzung und die spezifischen Umgebungsbedingungen für den Einsatz des jeweiligen Automatisierungssystems sowie unter Bezugnahme auf grundlegende

Qualitätskriterien für automatisierungstechnische Softwaresysteme (System-, Entwicklungs-, Betriebs-, Service- und Wartungsanforderungen). Sie erläutern, welche besonderen organisatorischen Rahmenbedingungen sich für die Entwicklung aus der Architektur ergeben, und bewerten die Qualität der Softwarearchitektur aus technischer und betriebswirtschaftlicher Sicht.

Allgemeine Informationen

Inhaltliche Voraussetzungen

PL	oder aus einem (naturwissenschaftlich-technischen) Bachelorstudium:
-undefined	- grundlegende Kenntnisse in (agilem) Projektmanagement

Kompetenzen

Kompetenz	Ausprägung
Gesellschaftliche Vertretbarkeit technischer Lösungen bewerten	Vermittelte Kompetenzen
Fachwissen erweitern und vertiefen und Lernfähigkeit demonstrieren	Vermittelte Kompetenzen
Komplexe Systeme und Prozesse analysieren, modellieren, realisieren, testen und bewerten	Vermittelte Kompetenzen
Probleme wissenschaftlich untersuchen und lösen, auch wenn sie unscharf, unvollständig oder widersprüchlich definiert sind	Vermittelte Kompetenzen
Anerkannte Methoden für wissenschaftliches Arbeiten beherrschen	Vermittelte Kompetenzen
Wissenschaftliche Ergebnisse und technische Zusammenhänge schriftlich und mündlich darstellen und verteidigen	Vermittelte Kompetenzen
Situations- und sachgerecht argumentieren	Vermittelte Kompetenzen
Projekte organisieren und im Team bearbeiten	Vermittelte Kompetenzen
Komplexe Aufgaben selbständig bearbeiten	Vermittelte Kompetenzen
Aufkommende Technologien einordnen und bewerten können	Vermittelte Kompetenzen
Sich selbst organisieren	Vermittelte Kompetenzen
Sprachliche und interkulturelle Fähigkeiten anwenden	Vermittelte Kompetenzen

Vorlesung / Übungen

Exemplarische inhaltliche Operationalisierung

Entscheidend für einen differenzierten Diskurs um die Qualität von und in Softwaresystemen sind eine allgemeine Definition des Qualitätsbegriffs bzw. der verschiedenen Qualitätsdimensionen für automatisierungstechnische Softwaresysteme sowie eine Diskussion um den betriebswirtschaftlichen Wert einer Software und den Komplexitätsbegriff. Darauf aufbauend können zweckmäßige Vorgehen zur additiven Entwicklung einer Softwarearchitektur am Beispiel wiederverwendbarer Softwareartefakte abgeleitet werden. Wiederverwendung kann am Beispiel von Mustern und Musterkatalogen (White-Box-Reuse) und Komponenten- bzw. Frameworkarchitekturen (Black-Box-Reuse) diskutiert werden. Als Beispiele professioneller Komponentenarchitekturen zum Aufbau verteilter technischer Softwaresysteme können Object Request Broker Architekturen wie CORBA bzw. das echtzeitfähige Open Source Derivat TAO (The ACE ORB), OPC/UA, die Funktionsbausteinarchitektur industrieller Leitsysteme (nach EN61499) aber auch integrierte Frameworks wie MS .NET herangezogen werden. Als weitere Abstraktionsstufe, also Architekturen mit begrenzter Intelligenz zur Verfolgung abstrahierter Zielvorgaben und zur Selbstrekonfiguration durch fortwährende Analyse des System- und Umgebungszustands, können Multiagentensysteme untersucht werden. Wesentlich für eine umfassende Beurteilung der Qualität von Softwarearchitekturen im Hinblick auf deren Einsatz unter spezifischen Bedingungen ist zudem ein grundlegender Diskurs über betriebswirtschaftlich, haftungsrechtlich und ethisch begründete Anforderungen.

Separate Prüfung

keine

Seminar

Exemplarische inhaltliche Operationalisierung

anspruchsvolle Seminarthemen können z. B. aus den Themengebieten
- wiederverwendbare Artefakte zum Aufbau der Architektur verteilter Softwaresysteme,
- professionelle Verteilungsarchitekturen,
- Multiagentensysteme,
- besondere betriebswirtschaftliche, haftungsrechtliche und ethische Anforderungen bei Softwaresystemen mit (verteilter) künstlicher Intelligenz und deren Auswirkungen auf die Gestaltung von Softwarearchitekturen

oder fachlich angrenzenden Themengebieten definiert werden.

Separate Prüfung

Benotet	Ja
Frequenz	Einmal im Jahr
Gewicht	50
Bestehen notwendig	Ja
Voraussetzung für Teilnahme an Modulprüfung	Ja

Prüfungskonzept

Auswertung wissenschaftlicher Literatur und Bewertung der Qualität von Softwarearchitekturen für verteilte Automatisierungssysteme im Hinblick auf vorgegebene automatisierungstechnische Fragestellungen.
Ergebnisvorstellung und wissenschaftlicher Diskurs in der Gesamtgruppe.

Projekt

Exemplarische inhaltliche Operationalisierung

Entwicklung einer Softwarekomponente für eine echtzeitfähige Verteilungsarchitektur in C++. Die Komplexität und der erwartete Arbeitsumfang zur Lösung der Aufgabenstellung richten sich nach dem verfügbaren Zeitkontingent des Projektteams (abh. von der Teamgröße).

Separate Prüfung

Benotet	Nein
Frequenz	Einmal im Jahr
Voraussetzung für Teilnahme an Modulprüfung	Ja

Prüfungskonzept

Entwicklung eines wiederverwendbaren Softwareartefakts für verteilte automatisierungstechnische Softwaresysteme. Ergebnisvorstellung und wissenschaftlicher Diskurs in der Gesamtgruppe.