

# Modulhandbuch 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

<b>Anerkannte Lehrveranstaltungen</b>	<u>SEKM Kreiser</u>
<b>Gültig ab</b>	Wintersemester 2020/21
<b>Dauer</b>	1 Semester
<b>ECTS</b>	5
<b>Zeugnistext (de)</b>	Qualitätsgesteuerter Entwurf komplexer Softwaresysteme
<b>Zeugnistext (en)</b>	Quality Controlled Development of Complex Software Systems
<b>Unterrichtssprache</b>	deutsch oder englisch
<b>abschließende Modulprüfung</b>	Ja

### Modulprüfung

**Benotet** Ja

---

**Konzept**

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.

---

**Frequenz**

Jedes Semester

## – Allgemeine Informationen

### Inhaltliche Voraussetzungen

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

### Handlungsfelder

Komplexe Rechner-, Kommunikations- und  
Eingebettete Systeme sowie komplexe Software-  
Systeme unter interdisziplinären Bedingungen  
entwerfen, realisieren und bewerten

---

Wissenschaftlich arbeiten und wissenschaftliche  
Erkenntnisse anwenden und erweitern

---

Fachliche Führungs- und Projektverantwortung  
übernehmen

---

undefined

### Learning Outcomes

<b>ID</b>	<b>Learning Outcome</b>
LO1	Studierende sind im Hinblick auf die Qualität eines Softwaresystems in der Lage: - zur vorhersagbaren, effizienten Entwicklung eines Softwaresystems bzw. einer Softwarearchitektur zielgerichtet angemessene Wiederverwendungsstrategien und professionelle Modellierungs- und Entwicklungswerkzeuge sowie den Rahmenbedingungen insgesamt angemessene Projektstrukturen einzusetzen. - die Softwarearchitektur für komplexe, verteilte Automatisierungssysteme unter Berücksichtigung der spezifischen Anforderungen hinsichtlich der besonderen Zielsetzung des jeweiligen Automatisierungssystems zu analysieren, zu konzipieren, zu entwerfen, zu implementieren, zu prüfen und zu bewerten. - die besonderen Anforderungen an die Servicequalität, an die Einsatzumgebung und die organisatorischen Rahmenbedingungen für die Entwicklung, die sich aus dem Entwicklungsprozess und einem angemessenen Lebenszyklusmanagement ergeben, zu erkennen und im Hinblick auf ihre Relevanz für die Softwarearchitektur des Automatisierungssystems zu analysieren und zu bewerten.

### Kompetenzen

<b>Kompetenz</b>	<b>Ausprägung</b>
Gesellschaftliche Vertretbarkeit technischer Lösungen bewerten	diese Kompetenz wird vermittelt

---

Fachwissen erweitern und vertiefen und Lernfähigkeit demonstrieren	diese Kompetenz wird vermittelt
--	---------------------------------

---

Komplexe Systeme und Prozesse analysieren, modellieren, realisieren, testen und bewerten	diese Kompetenz wird vermittelt
--	---------------------------------

---

Probleme wissenschaftlich untersuchen und lösen, auch wenn sie unscharf, unvollständig oder widersprüchlich definiert sind	diese Kompetenz wird vermittelt
--	---------------------------------

---

Anerkannte Methoden für wissenschaftliches Arbeiten beherrschen	diese Kompetenz wird vermittelt
---	---------------------------------

---

Wissenschaftliche Ergebnisse und technische Zusammenhänge schriftlich und mündlich darstellen und verteidigen	diese Kompetenz wird vermittelt
---	---------------------------------

---

Situations- und sachgerecht argumentieren	diese Kompetenz wird vermittelt
---	---------------------------------

---

Projekte organisieren und im Team bearbeiten	diese Kompetenz wird vermittelt
--	---------------------------------

---

Komplexe Aufgaben selbstständig bearbeiten	diese Kompetenz wird vermittelt
--	---------------------------------

---

Aufkommende Technologien einordnen und bewerten können	diese Kompetenz wird vermittelt
--	---------------------------------

---

Sich selbst organisieren	diese Kompetenz wird vermittelt
--------------------------	---------------------------------

---

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

---

## – Vorlesung / Übungen

<b>Typ</b>	Vorlesung / Übungen
------------	---------------------

<b>Separate Prüfung</b>	Nein
-------------------------	------

### **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.

## – Seminar

<b>Typ</b>	Seminar
------------	---------

<b>Separate Prüfung</b>	Ja
-----------------------------	----

### **Separate Prüfung**

<b>Benotet</b>	Ja
----------------	----

<b>Frequenz</b>	Einmal im Jahr
-----------------	----------------

<b>Gewicht</b>	50
----------------	----

<b>Bestehen notwendig</b>	Ja
-------------------------------	----

<b>Voraussetzung für Teilnahme an Modulprüfung</b>	Ja
--	----

**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.

**Konzept**

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**

**Typ** Projekt

---

**Separate Prüfung** Ja

---

**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

---

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