

 Hochschule Aalen	<b>Fakultät</b> Elektronik und Informatik	Modulbeschreibung <i>SPO 29 / SoSe</i> <span style="font-size: 2em; color: #00AEEF;">2015</span>
	<b>Studiengang</b> Informatik	
	<b>Modulkoordinator</b> Prof. Roy Oberhauser	

<b>Modul-Name</b>		Software Quality Management				<b>Modul-Nr : 57943</b>	
<b>CP</b>	<b>SWS</b>	<b>Workload</b>	<b>Kontaktzeit</b>	<b>Selbststudium</b>	<b>Angebot Beginn</b>	<b>Sem</b>	<b>Dauer</b>
10	8	300	120	180	<input checked="" type="checkbox"/> Wintersemester <input checked="" type="checkbox"/> Sommersemester	6 + 7	<input type="checkbox"/> 1 Semester <input checked="" type="checkbox"/> 2 Semester Semester
<b>Angestrebter Abschluss</b>		<b>Modultyp (PM/WPM/WM)</b>		<b>Studienabschnitt</b>		<b>Einsatz in Studiengängen</b>	
Bachelor of Science		PM - Pflichtmodul		HS - Hauptstudium		IN Schwerpunkt SE	
<b>Form der Wissensvermittlung</b>		<input checked="" type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Übung <input checked="" type="checkbox"/> Labor <input type="checkbox"/> Selbststudium <input type="checkbox"/> Seminar <input type="checkbox"/> Hausarbeit <input type="checkbox"/> Projektarbeit <input type="checkbox"/> Sonstiges: Referat, Bericht					

### Lernziele / Kompetenzen

#### **Fachkompetenz („Wissen und Verstehen“ und „Fertigkeiten“): Die Studierenden erwerben**

fortgeschrittene Softwareengineering Kenntnisse und wenden diese im Bezug auf Softwarearchitektur und Software Qualität an.

Studierende:

- kennen die Rolle und Aufgaben eines Softwarearchitekts, Architektur- Prozess, Konzepte, Prinzipien, Heuristiken, Methoden, und eine Vielzahl von Architektur- und Entwurfsmuster.
- können eine Softwarearchitektur erstellen, beschreiben, präsentieren, und bewerten.
- verstehen die Beeinflussung von plattformspezifischen Architektur, Middleware, und Applikation Frameworks auf die Softwarearchitektur und können Entscheidungen dazu begründen.
- kennen aktuelle Softwarearchitektur Themen wie Software Product Lines, Model-Driven Engineering, Cloud-Architektur, Service-Oriented Architecture, etc.
- kennen verschiedene gängige Qualitätsmanagement Ansätze und Maßnahmen und können geeignete Ansätze auswählen und begründen.
- kennen Qualitätsmodelle.
- kennen Requirements-Engineering Techniken und können diese anwenden.
- können Software Test Pläne entwerfen und Software Test Fälle erarbeiten.
- können eine Vielfalt von dynamische Testtechniken anwenden und die Testabdeckungsgrad messen.
- wissen wie die Effektivität von Entwicklungsprozessen bewertet werden können.
- kennen Code Qualitätsicherungstechniken und können Code Inspektionen durchführen.
- kennen den Einfluss von Design auf Qualität.
- können Statische Analyse durchführen und Aussagen über die Code Qualität sagen anhand verschiedene Metriken.
- üben Ihre Fähigkeit technischen und Verständnis Problemen bei (zum großen Teil Englische) Übungen zu überwinden.

#### **Überfachliche Kompetenz („Sozialkompetenz“ und „Selbstständigkeit“): Studierende**

- Können eine technische Präsentation in Englisch von Abstrakte Information (z.B. ein Entwurfsmuster oder Softwarearchitektur) selbstständig vorbereiten, darstellen, und präsentieren/erklären.
- Mehrsprachigkeit wird geübt auch durch Übungen, die zum großen Teil in Englisch beschrieben sind.

**Ggf. besondere Methodenkompetenz: -**

<b>Kompetenzbereich</b>	<b>Schwerpunkt</b>	<b>Teilschwerpunkt</b>	<b>In geringen Anteilen</b>
-------------------------	--------------------	------------------------	-----------------------------

<b>Fachkompetenz</b>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<b>Methodenkompetenz</b>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<b>Sozialkompetenz</b>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

**Lehrinhalte**

57631 Softwarearchitektur:

- Softwarearchitektur Qualitätseigenschaften und Architekturell-Überschneidende Aspekte
- Architektur Methodologien
- Architektur innerhalb SE Prozessen
- Architektur Repräsentation und Beschreibung
- Architekturellen Muster und Stile
- Abstraktion, Modellierung, und Entwurf an der Architektur Ebene (z.B. Model-Driven Development, ...)
- Plattform-spezifischen Architektur, Middleware, und Applikation Frameworks
- Produkt-Linie Architekturen
- Architektur Bewertung
- Aktuelle Architektur Themen

57631 Software- Qualitätsmanagement:

- Qualitätsmanagement Ansätze, Qualitätsmanagementsysteme
- Qualitätskriterien, Q-Normen und –Standards, Q-Modelle und Q-Prozesse, Qualitätsplanung und -kontrolle
- Vorbeugende, Konstruktive Q-Maßnahmen
- Analytische Q-Maßnahmen (Statische und Dynamische Prüfungen)
- Requirements-Engineering
- Qualitätspraktiken auf Code und Design Ebene
- Testingansätze
- Konfigurationsmanagement
- Metriken, Defektratenanalyse

<b>Zugangsvoraussetzung</b>	<p>Vorbereitung Teilnahme am Modul: Inhaltlich werden folgende Kenntnisse vorausgesetzt:</p> <ul style="list-style-type: none"> <li>- Software-Engineering (z.B. 57421)</li> <li>- Java und C# Kenntnisse (z.B. 57204, 57429)</li> </ul> <p>Prüfung: Voraussetzung zur Teilnahme an der Prüfungen sind die jeweils zugehörigen Übungsscheine.</p>
-----------------------------	---

**Enthaltene Teilmodule / Lehrveranstaltungen**

Fach-Nr.	Titel des Teilmoduls / Lehrveranstaltung	Lehrende	Art	SWS	CP	Sem	Teilmodulprüfung Art / Dauer / Benotung
57630	Software-Qualitätsmanagement	Prof. Roy Oberhauser	V Ü L	4	5	6	PLK 120 benotet
	<b>Teilmodultyp (PM/WPM/WM)</b>	<b>Studienabschnitt</b>	<b>Einsatz in Studiengängen</b>				
	PM - Pflichtveranstaltung	HS - Hauptstudium	IN Schwerpunkt SE				
Fach-Nr.	Titel des Teilmoduls / Lehrveranstaltung	Lehrende	Art	SWS	CP	Sem	
57631	Softwarearchitektur	Prof. Roy Oberhauser	V Ü L	4	5	7	PLK 120 benotet
	<b>Teilmodultyp (PM/WPM/WM)</b>	<b>Studienabschnitt</b>	<b>Einsatz in Studiengängen</b>				
	PM - Pflichtveranstaltung	HS - Hauptstudium	IN Schwerpunkt SE				

**Zugelassene Hilfsmittel**

	<p>Software-Qualitätsmanagement (57630): calculator, books, current course script, PDFs under Reference for the course in Moodle. Any notes must be original and in your own handwriting on A4 paper signed on each page with your signature and matrikel number in upper right.</p> <p>Softwarearchitektur (57631): books, current course script, PDFs under Reference for the course in Moodle. Any notes must be original and in your own handwriting on A4 paper signed on each page with your signature and matrikel number in upper right.</p>
--	--

<b>Sprache</b>	<input checked="" type="checkbox"/> Deutsch <input type="checkbox"/> Englisch <input type="checkbox"/> Spanisch <input type="checkbox"/> Französisch <input type="checkbox"/> Chinesisch <input type="checkbox"/> Portugiesisch <input type="checkbox"/> Russisch
----------------	--

<b>Literatur</b>	<p>Software Quality Management:</p> <ul style="list-style-type: none"> <li>- Andreas Spillner und Tilo Linz. Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified Tester; Foundation Level nach ISTQB-Standard. Oder Englisch: A Study Guide for the Certified Tester Exam. dpunkt-Verl.</li> <li>- Wallmüller, Ernest. Software Quality Engineering: Ein Leitfaden für bessere Software-Qualität. Hanser</li> <li>- McConnell, Steve: Code Complete. Microsoft Press.</li> <li>- Fowler, Martin: Refactoring. Wie Sie das Design vorhandener Software verbessern. Addison-Wesley.</li> <li>- Wallmüller, E.: Software-Qualitätsmanagement in der Praxis. Hanser Fachbuch.</li> <li>- Persse, J.: Process Improvement Essentials: CMMI, Six Sigma, and ISO 9001. O'Reilly Media, Inc.</li> <li>- Chrissis, Mary et al: CMMI : Guidelines for process integration and product improvement. Addison-Wesley.</li> <li>- Rupp, C., SOPHIST GROUP: Requirements-Engineering und –Management. Hanser Fachbuch.</li> <li>- Galin, D.: Software Quality Assurance. Pearson Education.</li> <li>- Sneed, H. M., Winter, M.: Testen objektorientierter Software. Hanser Fachbuch.</li> <li>- Ehrenberger, W.: Software-Verifikation. Hanser Fachbuch.</li> </ul> <p>Liggesmeyer, P.: Software-Qualität: Testen, Analysieren, und Verifizieren von Software. Spektrum Akademischer Verlag.</p> <ul style="list-style-type: none"> <li>- Wagner, Stefan: Software product quality control.</li> <li>- Grötter et al: The Developer's Guide to Debugging</li> <li>- Deval: Continuous Integration</li> <li>- Humble &amp; Farley: Continuous Delivery</li> </ul> <p>57631 Software Architecture:</p> <ul style="list-style-type: none"> <li>- Effektive Softwarearchitekturen: Ein praktischer Leitfaden von G. Starke</li> <li>- Designing Software Architectures: A Practical Approach by Cervantes &amp; Kazman</li> <li>- Software Architecture in Practice by Bass et al.</li> <li>- arc42 in Aktion von Starke &amp; Hruschka</li> <li>- Basiswissen für Softwarearchitekten Gharbi et al.</li> <li>- Design Patterns: Elements of Reusable Object-Oriented Software oder Entwurfsmuster. Elemente wiederverwendbarer objektorientierter Software von Gamma et al.</li> <li>- Head First Design Patterns oder Entwurfsmuster von Kopf bis Fuß von Freeman et al.</li> <li>- Entwurfsmuster: Das umfassende Handbuch von Geirhos</li> <li>- Pattern-Oriented Software Architecture: A System of Patterns by Buschmann, et al.</li> <li>- Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects, Vol. 2 by Schmidt et al.</li> <li>- Pattern-Oriented Software Architecture: Patterns for Resource Management</li> </ul>
------------------	---

	<p>Vol. 3 by Kircher &amp; Jain</p> <ul style="list-style-type: none"> <li>- The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise by Abbott and Fisher</li> <li>- Software Systems Architecture by Rozanski and Woods</li> <li>- Implementing Domain-Driven Design by Vaughn Vernon</li> <li>- Langlebige Software-Architekturen: Technische Schulden analysieren, begrenzen und abbauen von Lilienthal</li> <li>- Basiswissen Softwarearchitektur von T. Posch et al.</li> <li>- Just Enough Software Architecture: A Risk-Driven Approach by G. Fairbanks.</li> <li>- Practical Software Architecture: Moving from System Context to Deployment by T. Mitra</li> <li>- The Art of Software Architecture: Design Methods and Techniques by S. Albin</li> <li>- Documenting Software Architectures: Views and Beyond by P. Clements et al.</li> <li>- Service-Oriented Architecture (SOA): Concepts, Technology, and Design by T. Erl</li> <li>- SOA design patterns von T. Erl</li> <li>- Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions von Hohpe und Woolf</li> <li>- Cloud Architecture Patterns von B. Wilder</li> <li>- Refactorings in grossen Softwareprojekten. Komplexe Restrukturierungen erfolgreich durchführen von Roock &amp; Lippert</li> <li>- Domain-specific Languages by M. Fowler. Addison-Wesley</li> <li>- Evolving software systems von Mens et al.</li> <li>- Building Evolutionary Architectures by Ford et al.</li> <li>- Building Microservices von S. Newman</li> <li>- Microservice Architecture: Aligning Principles, Practices, and Culture by I. Nadareishvili et al.</li> <li>- Event-Driven Architecture: Softwarearchitektur für ereignisgesteuerte Geschäftsprozesse von Bruns &amp; Dunkel.</li> </ul>
<b>Zusammensetzung der Endnote</b>	gewichtet gemäß CP-Verteilung
<b>Bemerkungen / Sonstiges</b>	Die Teilmodule in dieses Modul sind auch vorgesehen als mögliche Wahlpflichtfächer für andere Studienschwerpunkte. Deshalb werden die Teilmodule einzeln geprüft.
<b>Letzte Aktualisierung</b>	2018-03-02 RO