

Verifikation komplexer SW-Systeme (VKS)

<i>Modulbezeichnung / Kürzel</i>	Verifikation komplexer SW-Systeme (VKS)	Stand: 22.05.2007
<i>Fachsemester</i>	2. Semester	
<i>Modulverantwortliche(r)</i>	Prof. Dr. Hans W. Nissen	
<i>Sprache</i>	Deutsch	
<i>Lehrformen / SWS</i>	Seminaristischer Unterricht mit Praktikum, V2, Ü1, P1	
<i>Kreditpunkte</i>	5 ECTS-Punkte	
<i>Arbeitsaufwand</i>	60 h Präsenz 90 Selbststudium	
<i>Voraussetzungen</i>	Grundlagen des Software Engineering, insbesondere Basistechniken zum Testen von Software	
<i>Lernziele/Kompetenzen</i>	<p>Die Studierenden besitzen fundierte Kenntnisse in fortgeschrittenen Testmethoden, ihren theoretischen Grundlagen und ihrer praktischen Anwendung.</p> <p>Die Studierenden kennen die Aufgaben und Ziele des Testmanagements. Sie kennen den grundsätzlichen Aufbau eines Testprozesses und können die Stärken und Schwächen eines konkreten Prozesses analysieren.</p> <p>Die Studierenden besitzen die Fähigkeit, ein Testkonzept für ein komplexes Software-System zu erstellen. Sie sind in der Lage, ein Fehlermanagement aufzubauen.</p> <p>Die Studierenden kennen die weit verbreiteten Testsprachen „UML 2 Testing Profile“ und TTCN-3. Sie sind in der Lage, Testfälle in diesen Sprachen zu spezifizieren. Die Studierenden kennen praxis-relevante Werkzeuge zur Unterstützung der verschiedenen Stufen der Software-Verifikation und besitzen praktische Erfahrung in ihrer Anwendung. Konkret besitzen sie Erfahrungen mit Werkzeugen für JUnit, TTCN-3 und den GUI-Test.</p> <p>Die Studierenden kennen die Ideen und Konzepte moderner Teststrategien, bei denen die Vorbereitung der Verifikation bereits im Systementwurf beginnt. Sie kennen sich weiterhin mit den Möglichkeiten und Grenzen der Testautomatisierung aus.</p>	
<i>Inhalt</i>	<p>Einleitung Vorgehensweisen bei der Software-Verifikation; Aufgaben und Ziele der Software-Verifikation</p> <p>Fortgeschrittene Testmethoden Prüfen von objektorientierter Software; Formale Techniken; Statische Codeanalyse; risikobasiertes Testen; Test einer GUI</p> <p>Testmanagement der fundamentale Testprozess (Planung, Überwachung, Steuerung); Testkonzept; Testreifegradmodelle (TMM); Testmetriken; Testauswertung; Fehlermanagement; Testmanagement-Werkzeuge</p> <p>Testsprachen Übersicht über Testsprachen; Aufbau eines Testsystems; abstrakte Testspezifikation (UML 2 Testing Profile); Systemtest-Spezifikation (TTCN-3)</p>	

	<p>Testwerkzeuge Übersicht über Werkzeugtypen; Entwicklungsorientiertes Testen (JUnit); Systemorientiertes Testen (TTCN-Werkzeug); GUI-Testen (autoit)</p> <p>Moderne Teststrategien Modellbasiertes Testen; Design for Testability; Test Driven Design; Testautomatisierung</p>
<i>Studien-/Prüfungsleistungen</i>	<p>mündliche Prüfung</p> <p>Voraussetzung: beständenes Praktikum</p>
<i>Medienformen</i>	<p>Tafel, Beamer, Computerdemonstration,</p> <p>Webseite: http://www.nt.fh-koeln.de/fachgebiete/inf/nissen/vks.html</p>
<i>Literatur</i>	<ul style="list-style-type: none"> •P. Liggesmeyer: Software-Qualität: Testen, Analysieren und Verifizieren von Software, Spektrum-Verlag, 2002. •A. Spillner, T. Roßner, M. Winter, T. Linz: Praxiswissen Softwaretest – Testmanagement, dpunkt-Verlag, 2006. •C. Willcock, T. Deiß, S. Tobies, S. Schulz, S. Keil, F. Engler: An Introduction to TTCN-3, John Wiley, 2005. •F. Westphal: Testgetriebene Entwicklung mit JUnit & FIT, dpunkt Verlag, 2005 •H. Balzert: Lehrbuch der Software-Technik, Band 2, Spektrum Akademischer Verlag, 1997.