

Modulbezeichnung	<b>Testautomatisierung</b>								
Modulverantwortliche(r)	Prof. Dr. J. Peleska								
Modulart	Pflicht/Wahl <input type="checkbox"/> Wahlpflicht <input checked="" type="checkbox"/>								
Spezialisierungsbereich	Automatisierung und Robotik, Systemsoftware / Eingebettete Systeme, Raumfahrt-Systemtechnik								
Dauer des Moduls	1 Semester								
Kreditpunkte	6 CP								
Arbeitsaufwand	<table style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2">Berechnung des Workloads</td> </tr> <tr> <td style="width: 80%;">Präsenz</td> <td style="text-align: right;">56 h</td> </tr> <tr> <td>Übungsbetrieb/Prüfungsvorbereitung</td> <td style="text-align: right;">124 h</td> </tr> <tr> <td style="border-top: 1px solid black;">Summe</td> <td style="text-align: right; border-top: 1px solid black;">180 h</td> </tr> </table>	Berechnung des Workloads		Präsenz	56 h	Übungsbetrieb/Prüfungsvorbereitung	124 h	Summe	180 h
Berechnung des Workloads									
Präsenz	56 h								
Übungsbetrieb/Prüfungsvorbereitung	124 h								
Summe	180 h								
Turnus des Moduls	i. d. R. angeboten alle 2 Jahre								
Voraussetzung für die Teilnahme	Keine <input type="checkbox"/> Folgende Inhaltliche Voraussetzungen: Grundlagen von Test und Verifikation								
Lehr- und Lernformen	Seminar <input type="checkbox"/> Vorlesung <input checked="" type="checkbox"/> Tutorium <input checked="" type="checkbox"/> Praktikum <input type="checkbox"/> Projekt <input type="checkbox"/>								
Lernziele	<p>Die Studierenden verfügen über ein vertieftes Verständnis für</p> <ul style="list-style-type: none"> <li>• Testfallentwurf</li> <li>• Bezug zwischen Anforderungen und Testfällen</li> <li>• Modell-basierte Testfallerzeugung</li> <li>• Algorithmen für die automatische Testfall-/Testdatenerzeugung</li> <li>• Äquivalenz zwischen erschöpfenden Tests und Korrektheitsbeweis</li> </ul>								
Lerninhalte	<ol style="list-style-type: none"> <li>1. Vorgehensmodelle und Testprozess</li> <li>2. Testarten auf unterschiedlichen Systemebenen</li> <li>3. Modell-basiertes Testen - die W-Methode von Chow</li> <li>4. Strukturelles Testen</li> <li>5. Modell-basiertes Testen von Echtzeitsystemen</li> <li>6. Spezialthemen aus den Gebieten <ul style="list-style-type: none"> <li>• SMT-Solver für die Berechnung konkreter Testdaten</li> <li>• Äquivalenzklassentests für nebenläufige Echtzeitsysteme</li> <li>• Überdeckungskriterien und ihr Bezug zum Korrektheitsbeweis</li> <li>• Mutationstests</li> </ul> </li> </ol>								
Prüfungsformen	i.d.R. Bearbeitung von Übungsaufgaben und Fachgespräch oder mündliche Prüfung								

Literatur

- R. Binder "Testing Object-Oriented Systems: Models, Patterns, and Tools", Addison-Wesley, 2000
- A. Spillner, T. Linz "Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified-Tester", dpunkt-Verlag, 2003.
- J. Peleska und M. Siegel "Test Automation of Safety-Critical Reactive Systems", South African Computer Journal, No. 19, pp. 53-77, 1997.
- J. Peleska "Formal Methods and the Development of Dependable Systems", Habilitationsschrift, Bericht Nr. 9612, Dezember 1996, Institut für Informatik und praktische Mathematik, Christian-Albrechts-Universität Kiel, 1997.
- Tsun S. Chow "Testing Software Design Modeled by Finite-State Machines", IEEE Transactions on Software Engineering, SE-4(3), pp. 178-186, März 1978.