

## TSW - Testen von Software

### TSW - Software Testing

---

<b>Allgemeine Informationen</b>	
<b>Modulkürzel oder Nummer</b>	TSW
<b>Modulverantwortlich(e)</b>	Prof. Dr. Lüsse, Jens (jens.luessem@fh-kiel.de) Prof. Dr. Woelk, Felix (felix.woelk@fh-kiel.de)
<b>Lehrperson(en)</b>	Dipl.-Inform. Hinkelmann, Kai (kai.hinkelmann@fh-kiel.de) Kasch, Henning (henning.kasch@fh-kiel.de) Prof. Dr. Lüsse, Jens (jens.luessem@fh-kiel.de)
<b>Wird angeboten zum</b>	Sommersemester 2020
<b>Moduldauer</b>	1 Fachsemester
<b>Angebotsfrequenz</b>	Regelmäßig
<b>Angebotsturnus</b>	In der Regel im Sommersemester
<b>Lehrsprache</b>	Deutsch
<b>Empfohlen für internationale Studierende</b>	Nein
<b>Ist als Wahlmodul auch für andere Studiengänge freigegeben (ggf. Interdisziplinäres Modulangebot - IDL)</b>	Nein

<b>Studiengänge und Art des Moduls (gemäß Prüfungsordnung)</b>
Studiengang: B.Eng. - E - Elektrotechnik Vertiefungsrichtung: Kommunikationstechnik und Embedded Systems Modulart: Wahlmodul Fachsemester: 6
Studiengang: B.Eng. - E - Elektrotechnik Vertiefungsrichtung: Technische Informatik Modulart: Wahlmodul Fachsemester: 6
Studiengang: B.Eng. - Me - Mechatronik Modulart: Wahlmodul Fachsemester: 4, 6
Studiengang: B.Eng. - Ming - Medieningenieur/-in Modulart: Pflichtmodul Fachsemester: 4
Studiengang: B.Eng. - Wing - Wirtschaftsingenieurwesen - Elektrotechnik Vertiefungsrichtung: Digitale Wirtschaft Modulart: Wahlmodul Fachsemester: 4, 6
Studiengang: B.Eng. - Wing - Wirtschaftsingenieurwesen - Elektrotechnik Vertiefungsrichtung: Kommunikationstechnik Modulart: Wahlmodul Fachsemester: 4, 6
Studiengang: B.Sc. - INI - Informationstechnologie Vertiefungsrichtung: Angewandte Informatik Modulart: Wahlmodul Fachsemester: 4, 6

## Kompetenzen / Lernergebnisse

*Kompetenzbereiche: Wissen und Verstehen; Einsatz, Anwendung und Erzeugung von Wissen; Kommunikation und Kooperation; Wissenschaftliches Selbstverständnis/Professionalität.*

**Fachkompetenz:**

Die Studierenden kennen die Prozesse der Qualitätssicherung von Software. Sie kennen die unterschiedlichen Testklassen, -Techniken und -Strategien. Sie kennen die wichtigsten Tools und deren Bedeutung in einem typischen Softwaretestprozess (Versionsmanagement, Continuous Integration Server, Test Frameworks, Code Metriken, ...).

**Methodenkompetenz:**

Sie können die jeweils richtige Strategie für eine bestimmte Testaufgabe auswählen und umsetzen.

Die Studierenden können die für die Testaufgabe geeigneten Tools auswählen und zielgerichtet einsetzen. Sie beschaffen sich Dokumentation über die eingesetzten Tools und eignen sich den Umgang mit den Tools eigenständig an. Sie können auf Basis von Anforderungsdokumenten Testfälle formulieren und durchführen.

## Angaben zum Inhalt

<b>Lehrinhalte</b>	<p>In diesem Modul werden die theoretischen Grundlagen des Testens vermittelt:</p> <ul style="list-style-type: none"> <li>• Testen im Software-Lebenszyklus</li> <li>• Testlevel / Testarten</li> <li>• Statische Techniken der Qualitätssicherung von Software</li> <li>• Dynamische Techniken der Qualitätssicherung von Software</li> <li>• Testorganisation</li> <li>• Softwarequalität und Risikomanagement</li> <li>• Testwerkzeuge</li> </ul> <p>Diese Grundlagen werden anhand praktischer Beispiele z.B. in Python vertieft:</p> <ul style="list-style-type: none"> <li>• Teststrategien: Random-, Differential-, Regression-Testing, ...</li> <li>• Testklassen: Unit-, Modul-, Komponenten-, Integrations-, GUI-, System- und Akzeptanz-Tests</li> <li>• Testtechniken: Fault Injection, Blackbox und Whitebox-Testing, Assertions</li> <li>• Testabdeckung / Code Coverage</li> <li>• BugTriage</li> </ul> <p>Das Modul vermittelt außerdem den praktischen Einstieg in typische Testtools, z.B.</p> <ul style="list-style-type: none"> <li>• Continuous Integration Server (Jenkins / Bamboo)</li> <li>• Unit Test Frameworks, z.B. Junit, PyUnit</li> <li>• Code Coverage Tools</li> <li>• Static Code Analyzers</li> <li>• Profiler</li> </ul>
<b>Literatur</b>	<ul style="list-style-type: none"> <li>• Agile Testing, M. Baumgartner et al, ISBN: 978-3-446-43194-2</li> <li>• Koomen, T., Pol, M. and Allott, S.K.: Test Process Improvement, Addison-Wesley Longman, 2002</li> <li>• Liggesmeyer, P.: Software-Qualität: Testen, Analysieren und Verifizieren von Software, Spektrum Verlag, 2009.</li> <li>• Spillner, A., Linz, T.: Basiswissen Softwaretest. Dpunkt-Verlag, 2012</li> </ul> <p>• Weitere Literatur wird rechtzeitig vor Beginn der Veranstaltung bekanntgegeben.</p>

<b>Lehrformen der Lehrveranstaltungen</b>	
<b>Lehrform</b>	<b>SWS</b>
Lehrvortrag	2
Labor	2

<b>Arbeitsaufwand</b>	
<b>Anzahl der SWS</b>	4 SWS
<b>Leistungspunkte</b>	5,00 Leistungspunkte
<b>Präsenzzeit</b>	48 Stunden
<b>Selbststudium</b>	102 Stunden

<b>Modulprüfungsleistung</b>	
<b>Voraussetzung für die Teilnahme an der Prüfung gemäß PO</b>	Keine
<b>TSW - Projektbezogene Arbeiten</b>	Prüfungsform: Projektbezogene Arbeiten Gewichtung: 50% wird angerechnet gem. § 11 Satz 3 PVO: Ja Benotet: Ja Anmerkung: In verschiedenen Tests wird im Labor der Umgang mit den einschlägigen Tools und die Fähigkeit sinnvolle Tests umzusetzen überprüft.
<b>TSW - Klausur</b>	Prüfungsform: Klausur Dauer: 120 Minuten Gewichtung: 50% wird angerechnet gem. § 11 Satz 3 PVO: Nein Benotet: Ja

<b>Sonstiges</b>	
<b>Empfohlene Voraussetzungen</b>	Fundierte Programmierkenntnisse aus den vorangegangenen Modulen (PRG, OOP, MOB).