Softwaretechnik *Software Engineering*								Modulnummer:
Bachelor Pflicht/Wahl ☒ Wahlpflicht ☐ Wahl ☐ Sonderfall ☐							Modulbereich: Pflicht	
Anzahl der SWS	V	UE	K	S	Prak.	Proj.	Σ	Kreditpunkte: 6	Turnus i. d. R. angeboten alle 2 Semester
	2	2	0	0	0	0	4		
Formale Voraussetzungen: -									
Inhaltliche Voraussetzungen: Software-Projekt									
Vorgesehenes Semester: ab 1. Semester									
Sprache: Deutsch									

Ziele: Die Studierenden verfügen über die folgenden fachlichen Kompetenzen:

- Methodenkompetenzen
- Analyse-/Design- und Realisierungskompetenzen
- Technologische Kompetenzen
- fortgeschrittene Methoden der Softwaretechnik kennen, beurteilen und umsetzen können
- Urteilsfähigkeit für technische Methoden
- Zusammenführung einzelner Methoden zu einem Ganzen

Die Studierenden verfügen über die folgenden sozialen Kompetenzen:

- Projektmanagement-Kompetenz zu Software-Projekten

Inhalte: Software-Metriken

- was ist eine Metrik?
- Messtheorie
- Skalen
- Prozess-, Produkt- und Ressourcenmetriken

Entwicklungsprozesse

- alternative Software-Entwicklungsprozesse (z.B. Clean-Room und Agile Entwicklung)
- Capability Maturity Model, Spice und Bootstrap
- Prozessverbesserungen
- Persönlicher Prozess

Software-Architektur

- Sichten und Blickwinkel, IEEE-Standard P1471
- Dokumentation von Software-Architektur und Architekturbeschreibungssprachen
- Entwurfs- und Architekturmuster und Referenzarchitekturen
- Qualitätseigenschaften
- Entwurf von Architekturen
- Analyse von Architekturen (insbesondere SAAM und ATAM)

Software-Produktlinien

- Definition und Beispiele
- Vor- und Nachteile
- Practice Areas
- Einführung von Produktlinien
- Ansätze zur technischen Realisierung
- Beschreibungen und Notationen (z.B. Feature-Graphen)
- Besonderheiten beim Requirementsengineering, Konfigurationsmanagement und Test
- Konfiguration von Produktlinien

Komponentenbasierte Entwicklung

- Eigenschaften, Vor- und Nachteile
- Komponentenmodell
- Schnittstellen und Kontrakte
- Managementfragen
- Rahmenwerke
- OMG CORBA und OMA
- Microsoft DCOM, OLE und ActiveX
- Sun Java und JavaBeans

Modellgetriebene Entwicklung

- Ideen, Eigenschaften, Vor- und Nachteile
- Werkzeugunterstützung (z.B.Eclipse Open Architecture Ware)

Kosten- und Aufwandsschätzung - insbesondere Function-Points und CoCoMo I und II

Empirische Softwaretechnik

- Bedeutung und Methoden der empirischen Softwaretechnik
- Bestandteile kontrollierter Experimente und Fallstudien

In der Vorlesung Softwaretechnik geht es um die Methodik der Software-Entwicklung nach Ingenieursprinzipien. Anhand der Projektsimulationssoftware SESAM kann die Durchführung eines Software-Projektes geübt werden. Das Kapitel ‘Empirische Softwaretechnik’ diskutiert grundlegende Methoden zum empirischwissenschaftlichen Erkenntnisgewinn bei der Softwareentwicklung.

Unterlagen (Skripte, Literatur, Programme usw.):

- Paul Clements und Linda Northrop: Software Product Lines: Practices and Patterns, Addison Wesley Professional, 2002
- Clemens Szyperski, Dominik Gruntz, Stephan Murer: Component Software, Addison Wesley Professional, 2002
- Norman E. Fenton, Shari L. Pfleeger: Software Metrics A Rigorous & Practical Approach, Second Edition, PWS Publishing Company, 1997
- Roger Pressman: Software Engineering – A Practioner's Approach, fünfte Ausgabe, McGraw-Hill, 2003
- Ludewig, Jochen; Lichter, Horst: Software Engineering Grundlagen, Menschen, Prozesse, Techniken. dpunkt.verlag, 2006
- Ian Sommerville: Software Engineering, Siebte Ausgabe, Addison-Wesley, 2004.
- Len Bass and Paul Clements and Rick Kazman: Software Architecture in Practice, zweite Auflage, Addison Wesley, 2003.
- Frank Buschmann, Regine Meunier, Hans Rohnert and Peter Sommerlad, Michael Stal: Pattern-oriented Software Architecture: A System of Patterns, Volume 1, Wiley, 1996.
- Christine Hofmeister, Robert Nord, Dilip Soni: Applied Software Architecture, Addison Wesley, Object Technology Series, 2000.
- Software Cost Estimation with COCOMO II; Barry W. Boehm et al.; Prentice Hall, 2000.
- Poensgen, Benjamin; Bock, Bertram: Die Function-Point-Analyse. Ein Praxishandbuch. Dpunkt Verlag, 2005. ISBN 978-3898643320
- Balzert, Helmut: Lehrbuch der Softwaretechnik Softwaremanagement. 2. Spektrum, Akademischer Verlag, 2008. ISBN 978-3-8274-1161-7
- Bunse, Christian ; Knethen, Antje von: Vorgehensmodelle kompakt. Spektrum-Akademischer Verlag, 2002. ISBN 978-3827412034
- Kruchten, Phillipe: The Rational Unified Process: An Introduction. Reading, Mass.: Addison-Wesley, 1998
- Beck, Kent: Extreme Programming Explained. Addison-Wesley, 2000 (The XP Series). ISBN 201-61641-6
- Kneuper 2006 Kneuper, Ralf: CMMI Verbesserung von Softwareprozessen mit Capability Maturity Model. 2. dpunkt.verlag, 2006. ISBN 3-89864-373-5
- Siviy, Jeannine M.; Penn, M. L.; Stoddard, Robert W.: CMMI and Six Sigma Partners in Process Improvement. Addison-Wesley, 2007 (SEI Series in Software Engineering). ISBN 978-0-321-51608-4
- Stahl, Thomas ; Volter, Markus ; Efftinge, Sven ; Haase, Arno: Modellgetriebene Softwareentwicklung Techniken, Engineering, Management. zweite Auflage. dpunkt.verlag, 2007
- Gamma, Erich ; Helm, Richard ; Johnson, Ralph ; Vlissides, John: Desig Patterns–Elements of Reusable Object-Oriented Software. Addison Wesley, 2003
- Pattern-oriented Software Architecture: A System of Patterns; Frank Buschmann, Regine Meunier, Hans Rohnert and Peter Sommerlad, Michael Stal; Volume 1, Wiley, 1996.
- Endres, Albert ; Rombach, Dieter: A Handbook of Software and Systems Engineering. Addison Wesley, 2003
- Prechelt 2001 Prechelt, Lutz: Kontrollierte Experimente in der Softwaretechnik Potenzial und Methodik. Springer, 2001
- Yin, Robert K.: Case Study Research. Bd. 5. SAGE Publications, 2003. ISBN 0-7619-2553-8

Form der Prüfung:
i.d.R. Bearbeitung von Übungsaufgaben und Fachgespräch oder mündliche Prüfung

Arbeitsaufwand	Präsenz	56 h
	Übungsbetrieb/Prüfungsvorbereitung	124 h
	Summe	180 h

Lehrende: Prof. Dr. R. Koschke	Verantwortlich: Prof. Dr. R. Koschke