

Software-Projekt-Vorlesung <i>Software Project (Lecture)</i>								Modulnummer: BA-901.01a		
Bachelor Pflicht/Wahl <input checked="" type="checkbox"/> Wahl <input type="checkbox"/> Basis <input type="checkbox"/> Ergänzung <input type="checkbox"/> Sonderfall <input type="checkbox"/>					Zugeordnet zu Masterprofil Sicherheit und Qualität (SQ) <input type="checkbox"/> KI, Kognition, Robotik (KIKR) <input type="checkbox"/> Digitale Medien und Interaktion (DMI) <input type="checkbox"/>					
Modulbereich: Projekte Modulteilbereich: (keine Angabe)										
Anzahl der SWS		V	UE	K	S	Prak.	Proj.	Σ	Kreditpunkte: 5	Turnus angeboten in jedem SoSe
		4	1	0	0	0	0	5		
Formale Voraussetzungen: Praktische Informatik 1										
Inhaltliche Voraussetzungen: -										
Vorgesehenes Semester: 2. Semester										
Sprache: Deutsch										
Kommentar: Für Informatik- und Wirtschaftsinformatik-Studierende ist diese Vorlesung Teil von Software-Projekt 1.										
Ziele: <ul style="list-style-type: none"> • Das angestrebte Ergebnis des Moduls insgesamt ist es, dass die Studierenden die methodischen und praktischen Fähigkeiten erwerben, in einer Gruppe eine Software-Lösung für ein vorgegebenes nicht-triviales Problem zu finden und zu realisieren. • Die zu erwerbenden fachlichen Kompetenzen umfassen alle notwendigen Aktivitäten in der Softwareentwicklung von der Anforderungsanalyse und Aufwandsschätzung, über den Architekturentwurf bis zur Implementierung und den Test. Ebenso gehören dazu die begleitenden Managementaspekte der Gruppenarbeit, Entwicklungsprozess, Planung, qualitätssichernde Maßnahmen, die Dokumentation und das Konfigurationsmanagement. • Die zu erwerbenden sozialen Kompetenzen betreffen das Projektmanagement in einem Software-Projekt sowie die Gruppenarbeit über einen längeren Zeitraum und die hierfür notwendige Selbstkompetenz (Zeitmanagement, Übernahme von Verantwortung und mehr). 										

Inhalte: Die folgenden, für ein solches Projekt notwendigen Themen der Softwaretechnik werden in der Vorlesung vermittelt (die Notation UML wird in den entsprechenden Abschnitten als Mittel zum Zweck und im methodischen Zusammenhang eingeführt):

Allgemeines

- was ist Software?
- Eigenschaften von Software
- Software-Lebenszyklus
- die besondere Bedeutung der Wartung und Evolution
- Softwarekrise
- was ist Softwaretechnik?

Projektplanung

- Grundbegriffe der Projektplanung
- Vorgehen bei der Planung
- Inhalt des Projektplans
- Gantt-Diagramme und kritischer Pfad
- Projektrisiken
- Softwareentwicklungsprozesse

Rechtlicher Rahmen der Softwareentwicklung

- Betriebsverfassungsgesetz (BetrVG), PersVG
- Arbeitsschutzgesetze, Verordnungen (BildscharbV)
- Datenschutzgesetze (BDSG)
- Normen und Richtlinien

Anforderungsanalyse

- Probleme bei der Anforderungsanalyse
- Schritte der Anforderungsanalyse
- Schritte der Ist-Analyse
- Erhebungstechniken bei der Ist-Analyse (Fragebögen, Interview im Kontext) und Soll-Analyse (Varianten des Prototypings)
- Aufbau und Inhalt der Anforderungsspezifikation
- Produktqualitäten
- Bedeutung und angestrebte Eigenschaften der Anforderungsspezifikation
- Regeln für die Anforderungsspezifikation
- Objektorientierte Anforderungsanalyse mit Anwendungsfällen, statischen und dynamischen Modellen mit Klassenbildung, die dem Liskovschen Substitutionsprinzip genügt (unter Verwendung der UML-Diagramme für Anwendungsfälle, Klassendiagramme, Interaktions- und Zustandsdiagramme)

Prüfung der Anforderungsspezifikation

- Software-Prüfungen im Allgemeinen
- Review-Varianten
- Abläufe von Reviews
- Review-Regeln
- Review-Checklisten
- Fallen und Gegenmittel

Software-Architektur

- Was ist Software-Architektur?
- Sichten (Views) und Blickwinkel (Viewpoints) der Software-Architektur
- Einflussfaktoren für die Software-Architektur
- Entwurf einer Software-Architektur
- Architekturstile
- Entwurfsmuster
- Modularisierung, Separation of Concern, Abstraktion, Information Hiding
- Architekturreview

Inhalte 2: Benutzungsschnittstellenentwurf

- Software-Ergonomie: Aspekte und Qualitäten
- Interaktionsformen und -mittel
- Werkzeuge
- Usability-Evaluationsverfahren

Einsatz von Datenbanken

- Aufgaben und Architektur von Datenbanksystemen; externe, konzeptionelle und interne Ebene
- Objektorientierte und relationale Datenbankmodellierung
- Abbildung von objektorientierten Schemata auf relationale Datenbankschemata
- Relationale Datenbanksysteme
- Structured Query Language (SQL): Schemadefinition, Datenmanipulation, Anfragen, Integritätsbedingungen
- Normalformen: 1NF, 2NF, 3NF

Implementierung

- Feinentwurf (Klassen, Zustands- und Aktivitätsdiagramme der UML)
- Programmiersprachen
- Programmierrichtlinien
- Code-Qualität und Metriken
- Vermeidung von Code-Redundanz
- Entwicklungsumgebungen

Test

- Möglichkeiten und Grenzen des Testens
- Testarten (Komponenten-/Integrations-/Systemtests)
- Test-Varianten: Black-Box, White-Box-Testen
- Testabdeckungsmaße
- Testvorbereitung, -durchführung und -protokollierung

Dokumentation

- interne Software-Dokumentation
- Benutzungshandbücher und Online-Hilfen

Änderungs- und Konfigurationsmanagement

- Wartung, Evolution und Reengineering
- Bedeutung der Software-Wartung
- Gesetze von Lehman
- Änderungsprozesse
- Werkzeuge für das Konfigurationsmanagement

Unterlagen (Skripte, Literatur, Programme usw.):

- R. Pressman: Software Engineering - A Practitioner's Approach. 6. Auflage, McGraw-Hill, 2004.
- I. Sommerville: Software Engineering. 8. Auflage, Addison-Wesley, 2006.
- W. Zuser, T. Grechenig, M. Köhle: Software Engineering mit UML und dem Unified Process. 2. Auflage, Pearson Studium, 2004.
- B. Brügge, A. H. Dutoit: Objektorientierte Softwaretechnik mit UML, Entwurfsmustern und Java. Pearson Studium, 2004.
- Jochen Ludewig, Horst Lichter: Software Engineering - Grundlagen, Menschen, Prozesse, Techniken. dpunkt.verlag, 2006.
- Helmut Balzert: Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering. 3. Auflage, Spektrum Akademischer Verlag, 2009.
- Helmut Balzert: Lehrbuch der Softwaretechnik: Softwaremanagement. 2. Auflage, Spektrum Akademischer Verlag, 2008.
- H. Störrle: UML 2 für Studenten. Pearson Studium, 2005.
- Chris Rupp, Stefan Queins, Barbara Zengler: UML 2 glasklar. 3. Auflage, Hanser Verlag, 2007.
- Chris Rupp: Requirements-Engineering und -Management. 5. Auflage, Hanser Verlag, 2009.
- Klaus Pohl, Chris Rupp: Basiswissen Requirements Engineering. dpunkt.Verlag, 2009.
- Klaus Pohl: Requirements Engineering - Grundlagen, Prinzipien, Techniken. 2. Auflage, dpunkt.Verlag, 2008.
- Ramez A. Elmasri, Shamkant B. Navathe: Grundlagen von Datenbanksystemen, 3. Auflage, Pearson Studium, 2009.

Form der Prüfung:

s. Software-Projekt 1

Arbeitsaufwand	Präsenz	70 h
	Übungsbetrieb/Prüfungsvorbereitung	80 h
	Summe	150 h

Lehrende:

Prof. Dr. R. Koschke, Dr. K. Hölscher

Verantwortlich:

Prof. Dr. R. Koschke