

LEHRVERANSTALTUNGSBESCHREIBUNG

1. Angaben zum Programm

1.1 Hochschuleinrichtung	Babes-Bolyai Universität
1.2 Fakultät	Mathematik und Informatik
1.3 Department	Informatik
1.4 Fachgebiet	Informatik
1.5 Studienform	Bachelor
1.6 Studiengang / Qualifikation	Informatik

2. Angaben zum Studienfach

2.1 LV-Bezeichnung	Software Engineering						
2.2 Lehrverantwortlicher – Vorlesung							
2.3 Lehrverantwortlicher – Seminar							
2.4 Studienjahr	2	2.5 Semester	2	2.6 Prüfungsform	P	2.7 Art der LV	Pflichtfach
2.8 Modulnummer	MLG5011						

3. Geschätzter Workload in Stunden

3.1 SWS	4	von denen: 3.2 Vorlesung	2	3.3 Seminar/Übung	2
3.4 Gesamte Stundenanzahl im Lehrplan	56	von denen: 3.5 Vorlesung	28	3.6 Seminar/Übung	28
Verteilung der Studienzeit:					Std.
Studium nach Handbücher, Kursbuch, Bibliographie und Mitschriften					14
Zusätzliche Vorbereitung in der Bibliothek, auf elektronischen Fachplattformen und durch Feldforschung					14
Vorbereitung von Seminaren/Übungen, Präsentationen, Referate, Portfolios und Essays					38
Tutorien					14
Prüfungen					14
Andere Tätigkeiten:					-
3.7 Gesamtstundenanzahl Selbststudium	94				
3.8 Gesamtstundenanzahl / Semester	150				
3.9 Leistungspunkte	6				

4. Voraussetzungen (falls zutreffend)

4.1 curricular	<ul style="list-style-type: none"> Grundlagen der Programmierung. Objektorientierte Programmierung
4.2 kompetenzbezogen	

5. Bedingungen (falls zutreffend)

5.1 zur Durchführung der Vorlesung	<ul style="list-style-type: none">• Videoprojektor. Internetzugang
5.2 zur Durchführung des Seminars / der Übung	<ul style="list-style-type: none">• Computers. UML Werkzeuge.

6. Spezifische erworbene Kompetenzen

Berufliche Kompetenzen	<ul style="list-style-type: none"> • die Identifikation der erforderlichen Methoden für die Softwareentwicklung • die Identifikation der erforderlichen Methoden für die Anforderungsanalyse der Softwaresysteme • die Verwendung von Methoden und Techniken der Anforderungsanalyse und der Programmierwerkzeugen für Softwaresysteme • die Verwendung der Methoden zur Überprüfung der Softwareprogramme • die Entwicklung der informatischen Projekte
Transversale Kompetenzen	<ul style="list-style-type: none"> • die Erweiterung der Kommunikationsfähigkeiten • das Erlangen der Fertigkeit Aufgaben unter der Leitung eines Teammanagers zu lösen • unternehmerischen Fähigkeiten. Eigeninitiative

7. Ziele (entsprechend der erworbenen Kompetenzen)

7.1 Allgemeine Ziele der Lehrveranstaltung	<ul style="list-style-type: none"> • das Aneignen der Kenntnissen zur Softwareentwicklung
7.2 Spezifische Ziele der Lehrveranstaltung	<ul style="list-style-type: none"> • das Aneignen der Kenntnissen über Methodologien der Softwareentwicklung • Die Vertrautheit mit Konzepten der modernen Softwareentwicklung • Kenntnisse der Modelltypen und Werkzeuge zur Systemmodellierung • das Verständnis der Wichtigkeit von Softwaredokumentation in alle Etappe des Lebenszyklus

8. Inhalt

8.1 Vorlesung	Lehr- und Lernmethode	Anmerkungen
1. Organisatorische Diskussionen. Lebenszyklus eines Softwareprodukts. Softwaretechnik: Evolution. Definitionen. Das Wasserfallmodell. Das Spiralmodell. Inkrementelles Vorgehensmodell. Extreme Programming	Vortrag, Gespräch, Fallstudien	
2. Problemlösen und Softwareentwicklung. Prinzipien der Kommunikation.	Vortrag, Gespräch, Fallstudien	
3. Systemmodellierung. Anwendungsdomäne. Werkzeuge zur Systemmodellierung	Vortrag, Gespräch, Fallstudien	
4. Überblick über UML. Anforderungsanalyse. Modellierung von Klassen.	Vortrag, Gespräch, Fallstudien	
5. UML. Dynamische Modellierung. Architekturmodellierung	Vortrag, Gespräch, Fallstudien	
6. Anforderungsanalyse	Vortrag, Gespräch, Fallstudien	
7. Statische Modellierung. Dynamische	Vortrag, Gespräch,	

Modellierung. Grundlagen der Modellierung. Entwurfsetappen	Fallstudien	
8. Architektur und Abhängigkeitsbeziehungen. Prinzipien der Architekturmodellierung	Vortrag, Gespräch, Fallstudien	
9. Architekturmodellierung. Datenmodellierung	Vortrag, Gespräch, Fallstudien	
10. Die logische Systemarchitektur. Etappen der objektorientierte Modellierung	Vortrag, Gespräch, Fallstudien	
11. Interfacedesign	Vortrag, Gespräch, Fallstudien	
12. Mensch-Computer-Interaktion	Vortrag, Gespräch, Fallstudien	
13. Methoden beim Testen von Software	Vortrag, Gespräch, Fallstudien	
14. Der Faktor Mensch in der Softwareentwicklung	Vortrag, Gespräch, Fallstudien	
Literatur		
<p>1. FLAATEN, P.O., MCCUBBREY, D.J., O'RIORDAN, P.D., BURGESS, K.: Foundations of Business Systems, Dryden Press, 1st ed. 1989, 2nd ed. 1997.</p> <p>2. FOWLER, M., SCOTT, K.: UML Distilled: A Brief Guide to the Standard Object Modeling Language, Addison-Wesley, 2nd ed., 1999.</p> <p>3. IACOBSON, I., BOOCH, G., RUMBAUGH, J.: The Unified Software Development Process, Addison-Wesley, 1999.</p> <p>4. MARTIN, R.C.: Agile Software Development: Principles, Patterns, and Practices, Prentice Hall, 2002.</p> <p>5. PÂRV, B.: Analiza si proiectarea sistemelor, Univ. Babes-Bolyai, CFCID, Facultatea de Matematica si Informatica, Cluj-Napoca, 2002, 2003, 2004.</p> <p>6. PRESSMAN, R.S.: Software Engineering - A Practitioners Approach, McGraw-Hill, 3rd ed. 1992; 4th ed. 1996, 5th ed. 2001, 6th ed. 2005.</p> <p>7. SCHACH, S.R.: Object-Oriented and Classical Software Engineering, McGraw-Hill, 5th ed., 2002, 6th ed. 2005.</p> <p>8. SOMMERVILLE, I.: Software Engineering, Addison-Wesley, 5th ed. 1996, 6th ed. 2000, 7th ed. 2004</p>		
8.2 Seminar / Übung	Lehr- und Lernmethode	Anmerkungen
S1. Fallstudie: Anforderungsanalyse (1) - Grundanforderungen	Rückschau, Erklärungen, Beispiele	2 Stunden jeden 2 Wochen
S2. Fallstudie: Anforderungsanalyse (2) - modellierung von Klassen	Rückschau, Erklärungen, Beispiele	
S3. Prüfung: Anforderungsanalyse (3) – Anwendungsfall-Modellierung. Sequenzdiagramme	Rückschau, Erklärungen, Beispiele	
S4. Prüfung 1: Objektorientierte Modellierung	Rückschau, Erklärungen, Beispiele	
S5. Fallstudie: Entwurf (1) - Anwendungsfälle	Rückschau, Erklärungen, Beispiele	
S6. Prüfung 2: Entwurf (2). GRASP	Rückschau, Erklärungen, Beispiele	
S7. Fallstudien	Rückschau, Erklärungen, Beispiele	

L1. Organisatorische Diskussionen	Erklärungen, Beispiele	2 Stunden jeden 2 Wochen
L2. Start des Projekts (Etappe 1)	Erklärungen, Beispiele	
L3. Projekt (Etappe 1) Bildung	Erklärungen, Beispiele	
L4. Projekt (Etappe 1) abgeben. Start des Projekts (Etappe 2)	Erklärungen, Beispiele	
L5. Projekt (Etappe 2) Bildung	Erklärungen, Beispiele	
L6. Projekt (Etappe 2) abgeben. Start des Projekts (Etappe 3)	Erklärungen, Beispiele	
L7. Projekt (Etappe 3) abgeben	Erklärungen, Beispiele	

Literatur

1. Object Mentor Homepage, [<http://www.objectmentor.com>].
2. Agile Modeling Homepage, [<http://www.agilemodeling.com>].
3. Software Engineering Body of Knowledge, IEEE, 2004. [<http://www.swebok.org>].

9. Verbindung der Inhalte mit den Erwartungen der Wissensgemeinschaft, der Berufsverbände und der für den Fachbereich repräsentativen Arbeitgeber

- Der Kurs folgt die IEEE und ACM Curricula Empfehlungen für das Informatikstudium.
- Der Kurs existiert in der Mehrzahl der rumänischen und ausländischen Universitäten.
- Die Softwarefirmen finden den Kursinhalt zehr wichtig für die Ausbildung der Zukünftigen Softwareentwickler.

10. Prüfungsform

Veranstaltungsart	10.1 Evaluationskriterien	10.2 Evaluationsmethoden	10.3 Anteil an der Gesamtnote
10.4 Vorlesung	Kenntnisse der im Kurs behandelten Themen	2 Tests Midterm-Prüfung Prüfung	10% 20% 25%
10.5 Seminar / Übung	Die Fähigkeit Modellierungstechniken für das Lösen konkreter Probleme einzusetzen	3 Mini-projekte der Besuch des Kurses	45% 10%
10.6 Minimale Leistungsstandards			
<ul style="list-style-type: none"> • Note 5 auf einer Skala von 1 bis 10. 			

Ausgefüllt am:

Vorlesungsverantwortlicher

Seminarverantwortlicher

Genehmigt im Department am:

Departmentdirektor