

SYLLABUS
pentru disciplina:

“Proiectarea arhitecturala a sistemelor software”

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DOMENIUL /SPECIALIZAREA INFORMATICĂ /INFORMATICĂ APLICATĂ -
Master în Informatică

Anul de studii: _____ 2 _____

Semestrul (1 sau 2) 1

Titularul cursului: Conf.dr.ing. Ioana Șora					
Colaboratori:					
Numar de ore/saptamana/Verificarea/Credite					
Curs	Seminar	Laborator	Proiect	Examinare	Credite
28	0	0	21	E	8

A. OBIECTIVELE CURSULUI

Analiza si proiectarea sistemelor software complexe, utilizarea stilurilor si tiparelor arhitecturale adecvate atributelor de calitate vizate.

B. SUBIECTELE CURSULUI

PARTEA 1: FUNDAMENTE: 1.1.Concepte : Arhitectura software, Stiluri arhitecturale, Tipare arhitecturale (architectural patterns), Tipare de proiectare (design patterns), Frameworks, Reference architectures. **1.2.Architecture Business Cycle.** [2 ore]

PARTEA 2: METODE DE PROIECTARE ARHITECTURALA: 2.1. Calitati si tactici arhitecturale: Atribute de calitate (availability, modifiability, performance, security, testability, and usability); Tactici arhitecturale pentru obtinerea atributelor de calitate; Relatia dintre tactici arhitecturale si tipare arhitecturale; Studiu de caz. **2.2. Stiluri si tipare arhitecturale (Architectural Styles and Patterns): 2.2.1.Stiluri si tipare arhitecturale fundamentale:** Layers; Pipe-and-Filters; Blackboard; Event-Bus. **2.2.2. Stiluri si tipare arhitecturale pentru sisteme distribuite:** Aplicatii distribuite (Client-Server, Peer-to-peer, N-Tiers); Middleware (Broker); Studiu de caz. **2.2.3. Stiluri si tipare arhitecturale pentru sisteme adaptive:** Reflection; Microkernel. **2.3. Proiectarea arhitecturala ghidata de atributele de calitate (ADD – Attribute Driven Design)** [15]

PARTEA 3: STRUCTURI SI TIPURI DE VEDERI ARHITECTURALE (ARCHITECTURAL VIEWTYPES). DOCUMENTAREA ACESTORA: 3.1. Tipuri de vederi statice (module viewtypes: decomposition, uses, generalization); 3.2. Tipuri de vederi dinamice (component-and-connector viewtypes: data stream, call-return, publish-subscribe, concurrency); 3.3. Alegerea tipurilor de vederi relevante scopurilor; 3.4. Documentarea tipurilor de vederi arhitecturale (UML, ADL's, standarde-recomandari). [6 ore]

PARTEA 4: REUTILIZAREA LA NIVEL ARHITECTURAL: 4.1. Familii de produse software – Software Product Lines; 4.2. Proiectarea cu componente software; tehnologii de componente. 4.3. Studiu de caz. [5 ore]

C. SUBIECTELE APLICATIILOR (proiect)

Se va proiecta arhitectura unui sistem, urmarind asigurarea atributelor de calitate dorite ale acestuia, prin metoda ADD. Se vor utiliza stiluri si tipare arhitecturale studiate la curs. Se vor documenta vederile arhitecturale relevante ale proiectului. [21 ore]

D. BIBLIOGRAFIE Se indică maximum trei titluri bibliografice de referință

1. L. Bass, P. Clements, R. Kazman: *Software Architecture in Practice, Second Edition*, Addison Wesley Professional, 2003.
2. P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, J. Stafford: *Documenting Software Architectures: Views and Beyond*, Addison-Wesley Professional, 2002

3. F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal: *Pattern-Oriented Software Architecture, Volume 1: A System of Patterns*, Wiley, 1996.

E. PROCEDURA DE EVALUARE

Examen scris, pondere 50% in nota finala. Proiectul are o pondere de 50% in nota finala.

F.COMPATIBILITATE INTERNATIONALA

University of Groeningen, Olanda.

University of Utrecht, Olanda

University of Minnesota, USA.

Data: 25.03.2008

DIRECTOR/SEF DEPARTAMENT/CATEDRA

TITULAR DE DISCIPLINĂ,