

Enginyeria del Software I

Codi	Tipus	Curs/semestre	Crèdits ECTS
21290	• Troncal • Semestral	3er curs / 5è semestre	5

Professors:

<i>Nom</i>	<i>Dpt/Unitat</i>	<i>Despatx</i>	<i>Adreça e-mail</i>	<i>Telèfon</i>
Xavier Roca	Ciències de la Computació	QC/1040	Xavier.Roca@uab.cat	93 581 2578
Josep Lladós	Ciències de la Computació	QC/1052	Josep.Lladós@uab.cat	93 581 2403
Josep M. Gonfaus Javier Mateo Partha Pratim David Fernández Marc Castelló Mireia Bellot Elena Cerón	Ciències de la Computació	QC/1046		

Objectius:

Un primer objectiu és donar una visió global i ordenada del procés de desenvolupament del software que no sigui simplement el de la programació --que constitueix, només, una fase dins de tot el procés de la enginyeria del software.

El temari se centra en les dues primeres etapes del cicle de vida del software: anàlisi i disseny. Per cada una de aquestes dues grans parts del curs estudiarem en detall dues metodologies: estructurada i orientada a l'objecte. La part central i més important del curs estarà dedicada a la metodologia orientada a objecte i, més concretament, a UML com a llenguatge de modelat i RUP com a paradigma de desenvolupament. Tot això es complementarà en casos pràctics utilitzant l'eina Rational Rose.

L'objectiu és que l'alumne sigui capaç d'utilitzar aquestes metodologies per tal de realitzar l'anàlisi i el disseny de problemes reals, que li permetrà abordar la creació de software de manera rigorosa.

Coneixements

- Entendre què és el software, quins són els objectius que ens plantejem al desenvolupar un software i la problemàtica del associada al seu desenvolupament.
- Conèixer els principals paradigmes de desenvolupament del software.
- Aprendre què és l'anàlisi de requeriments i perquè és important. Abstracció i partició
- Aprendre què és el disseny. Com aconseguir un bon disseny. Modularitat efectiva.
- Introducció al d'UML: notació i construcció dels diagrames.
- Introducció a un procés de desenvolupament del software: RUP.

Habilitats

- A partir d'un problema obert enunciat en llenguatge natural, saber analitzar que ens esta demanant el client: crear un document d'especificació.
- A partir de la informació obtinguda en el document d'especificació modelar-ho a partir dels diferents diagrames de UML.
- Se capaç de veure que alhora de modelar hi ha punts que no disposem de tota la informació i per tant s'ha de tornar a analitzar el problema: Iterar.
- Portar a terme aquestes activitats en un context orientada a objecte: els models (diagrames) que obtenim son diferents punts de vista del mateix problema on cada un modela algun característica de les metodologies orientades a objecte.
- Consciència de que un Enginyer en Informàtica no és només un programador i de que desenvolupar software és molt més que programar.
- Saber buscar i seleccionar la informació necessària per assolir els objectius plantejats.

Competències

- Treball en equip.
- Capacitat d'anàlisi i síntesi.
- Capacitat de resolució de problemes.
- Raonament crític.
- Motivació per la qualitat.
- Capacitat de resolució de problemes.
- Comunicació oral i escrita.
- Capacitat d'organització i planificació.
- Aprenentatge autònom.
- Lideratge

Capacitats prèvies:

Es recomana que l'alumne tingui coneixements i capacitats de:

- Coneixement de Programació.
- Coneixement dels principis bàsics dels llenguatges orientats a objecte.
- Estructures de dades bàsiques.

Aquests conceptes corresponen a continguts de les assignatures:

- Algorismes i programació
- Llenguatges de programació.
- Estructures de dades.

(T:teoria, S:seminaris, PS:preparació de seminaris, L:laboratori tancat, PP:laboratori obert, E:estudi, AA:altres activitats)

1. Principis de l'Enginyeria del Software.	T	S	PS	L	E	PP	AA	Total
	2				1,5			3,5

- **Definició i objectius de l'ES.** Definició de software. Característiques del software. Aplicacions del software. Definició d'ES. Objectius de l'ES.
- **Evolució del software.** Etapes. Crisi del software: problemes i causes.
- **Procés, mètode i eina.** Definicions. Activitats en el procés de desenvolupament del software.
- **Paradigmes del desenvolupament del software.** Model lineal seqüencial (cicle de vida clàssic). Model de prototipat. Model evolutiu. Model en espiral.

2. Anàlisi de Requeriments del Software.	T	S	PS	L	E	PP	AA	Total
	2	2	7		4			15

- **Introducció.** Tipus de requeriments. Tasques a realitzar.
- **Comprensió del problema.** Tècniques de comunicació. Problemes associats. Principis de l'anàlisi.
- **Especificació de requeriments.** Propietats desitjables d'una ER. Estàndards d'ES. Revisió i validació de l'especificació.

3. Disseny del Software.	T	S	PS	L	E	PP	AA	Total
	2	0,5	2,5		3,5			8,5

- **Introducció.** Procés de disseny. Disseny de dades, disseny arquitectònic, disseny de la interfície disseny procedimental. Principis (objectius) del disseny.
- **Conceptes del disseny.** Abstracció. Modularitat. Refinament.
- **Disseny modular efectiu.** Independència funcional. Cohesió Acoblament. Heurístiques per a un disseny modular efectiu.

4. Software Orientat a Objectes	T	S	PS	L	E	PP	AA	Total
	1				1			2

- **Introducció.**
- **Principis.** Abstracció, encapsulament, modularitat, jerarquia.
- **Conceptes.** Classe, objecte, atribut, operació, interfície, component, paquet, subsistema, relacions.

5. El Llenguatge Unificat de Modelat (UML)	T	S	PS	L	E	PP	AA	Total
	3	5	15		9	16		48

- **Introducció.** Visió general d'UML. Notació.
- **Model estructural.** Classes. Relacions. Mecanismes comuns. Diagrames. Diagrames de classes.
- **Model de comportament.** Interaccions. Casos d'ús. Diagrames de casos d'ús. Diagrames d'interacció. Diagrames d'activitats. Diagrames d'estats.

- **Model arquitectònic.** Diagrames de components. Diagrames de desplegament.
- **Exemples.**

6. El Procés Unificat de Rational (RUP)	T	S	PS	L	E	PP	AA	Total
	3	2	4		8	6		23
• Introducció. • Les 6 “best practices”. • Què és RUP? • Estructura estàtica del RUP. • Estructura dinàmica del RUP.								

Preparació de l'avaluació final i del portafoli de l'assignatura	T	S	PS	L	E	PP	AA	Total
							24	24
Inclou la presentació del cas pràctic (problema guia), preparació i lliurament de la memòria, controls i la preparació de l'avaluació final de l'assignatura.								

Metodologia docent:

L'assignatura d'Enginyeria del Software 1 té una forta component aplicada. Per aquest motiu, després d'introduir els conceptes teòrics fonamentals, l'assignatura es desenvolupa segons una metodologia ABP (Aprenentatge Basat en Problemes). D'aquesta manera, l'alumne adquirirà els coneixements a partir d'un cas pràctic (problema guia) de dimensió real. El problema guia dirigirà el procés d'aprenentatge al llarg del curs, de manera que el treball pràctic o els seminaris s'estructuraran entorn d'aquest cas. L'alumne haurà d'adquirir el perfil d'analista / arquitecte del software per prendre les decisions oportunes que concloguin en la construcció d'una proposta de disseny del sistema proposat. El professor farà el paper de tutor i, en situacions determinades, de client, fomentant simulacions de diàleg desenvolupador – client.

TEORIA:

Consisteix en classes magistrals amb material multimèdia disponible al Campus Virtual de la UAB. L'objectiu principal d'aquestes classes és introduir les nocions bàsiques sobre els processos, mètodes i eines relatius a la visió del cicle de vida del software des d'un punt de vista d'enginyeria. S'exposarà quins són els principis i paradigmes de l'enginyeria del software. Ateses les tendències actuals de desenvolupament de software, es posarà atenció en els models orientat a objecte i en els processos iteratiu i incremental. Després d'un primer bloc introductori, es treballarà en les dues primeres etapes del cicle de vida del software: anàlisi de requeriments i disseny. Les classes de teoria han de permetre a l'alumne agafar una visió real del rigor, planificació i la sistemàtica que requereix el desenvolupament professional del software. Les classes de teoria es duran a terme en les 6 primeres setmanes de curs a raó de 3 hores presencials per setmana.

SEMINARIS:

Els seminaris són classes amb un nombre reduït d'alumnes. Es poden considerar com l'eix conductor de l'assignatura, i tenen una doble funció, enllaçant per una banda amb les classes magistrals i per l'altra amb les sessions pràctiques de laboratori. Per una banda, els seminaris complementaran els continguts tecnològics exposats a les classes magistrals, completant-ne la seva comprensió. Per l'altra seran el marc de treball on debatre, establir les bases i desenvolupar el problema guia. Se li donaran a l'estudiant els coneixements necessaris per tirar endavant el treball pràctic o se li indicarà on aconseguir-los. En els

seminaris es promou fonamentalment la capacitat d'anàlisi i síntesi, així com el raonament crític i la presa de decisions de l'alumne front a la resolució del problema guia.

S'ha programat 10 seminaris al llarg del curs (setmana 7 a setmana 16), de màxim una hora presencial cadascun. Els seminaris estan agrupats per activitats (cada activitat està dissenyada per treballar en una iteració del cas guia). En total hi ha 4 activitats (iteracions del problema guia). Cada activitat seguirà el cicle *anàlisi del problema – cerca d'informació – posada en comú / discussió – preparació de solució – presentació*. El treball dels seminaris a més d'adquirir coneixements sobre els aspectes concrets que plantegi, servirà per avançar en el treball del problema guia. El darrer seminari de cada iteració serà també d'acceptació/avaluació dels materials a lliurar.

LABORATORIS:

En l'àmbit professional el desenvolupament del software es du a terme amb l'ajut d'eines CASE (Computer Aided Software Engineering). Una eina CASE és una eina informàtica d'ajut a l'enginyer del software en les diverses etapes de desenvolupament. Pot ser una base de dades de recollida de requeriments, un editor gràfic per a la programació visual, una eina per monitoritzar l'execució de l'aplicació en l'etapa de proves, un gestor de la configuració, etc. En el cas d'ES1, s'utilitzarà una eina per dur a terme el problema guia en les seves etapes d'anàlisi i disseny. En particular, les eines de la plataforma d'IBM-Rational. Els alumnes tindran a la seva disposició aquesta eina en els laboratoris d'informàtica per utilitzar-la com a suport en el desenvolupament del cas guia. En els materials de l'assignatura es lliurarà als alumnes exemples i tutorials d'utilització de l'eina. Durant els seminaris també es resoldran qüestions sobre l'utilització de l'eina. No hi haurà sessions presencials de laboratori.

ORGANITZACIÓ:

Cada **grup de classe** (aproximadament 75 alumnes matriculats) es divideix en **equips de treball** de 5 persones que treballaran conjuntament en el decurs del curs (seminaris i cas pràctic). Les sessions de **teoria** seran sessions plenàries a les que assistiran tots els alumnes (3 hores les setmanes en que es programi teoria, un total de 6 setmanes). Els **seminaris** es faran per equips de treball, de manera que les setmanes en que s'hagi programat seminari, cada equip assistirà a una sessió presencial d'una hora màxim d'entrevista amb el professor.

El quadre que es presenta al final d'aquesta secció il·lustra la distribució de sessions al llarg del curs i els seus continguts.

TEORIA								CAS GUIA (AVALUACIÓ CONTINUADA)							
									Inici				Elaboració	Constr 2a Iter	
1	2	3	4	5	6	7	8		9	10	11	12	13	14	15
21S-25S	28S-20	50-90	120-160	190-230	260-300	2N-6N	9N-13N		16N-20N	23N-27N	30N-4D	7D-11D	14D-18D	11G-15G	18G-22G
								Modelat del Negoci							
Intro ES Anàlisi Disseny Anàlisi Disseny UML UML RUP RUP Problema Avaluació								Model de Casos d'Ús del Negoci i Model d'Objectes del Negoci							
								Requisits							
								Visió							
								Model de Casos d'Ús					✓		
								Especificació de Casos d'Ús					✓		
								Anàlisi / Disseny							
								Model d'Anàlisi / Disseny						✓	✓
								Model de Dades							
								Implementació							
								Prototipus d'Interfícies d'Usuari							
								Proves							
								Casos de Proves Funcionals							
								Desplegament							
								Model de Desplegament							
								Gestió de Canvis i Configuració							
								Gestió del projecte							
								Pla de Desenvolupament del Software en versió 1.0							

Avaluació:

L'objectiu del procés d'avaluació és verificar que l'alumne ha assolit els coneixements i habilitats definits en els objectius de l'assignatura, així com les competències, aquests últimes en un grau compatible amb la situació d'aquesta assignatura en el cinquè quadrimestre de la titulació.

ACTIVITATS I INSTRUMENTS D'AVALUACIÓ:

L'avaluació es durà a terme en base al treball pràctic desenvolupat per l'alumne i inclòs en el seu portfoli (resolució final de problema guia, material generat en les activitats dels seminaris), el grau d'implicació en els seminaris i els coneixements tecnològics assolits de la matèria a estudi. Es contemplaran els següents instruments d'avaluació:

- **Avaluació individual (AI).** Prova escrita presencial per valorar el grau de coneixements assolits per l'alumne a nivell individual. Es realitzarà en l'hora de teoria de la **6a setmana de curs**.
- **Avaluació col·lectiva** dels equips de treball (**AC**). Prova del rendiment conjunt dels diferents grups de treball per resoldre el problema guia i les diferents activitats presentades als seminaris. Consta de dos instruments:
 - o **AC1 (avaluació tècnica).** Documentació lliurada pels alumnes del seu treball pràctic. S'avaluarà la resolució tècnica del problema guia.
 - o **AC2 (avaluació de progrés i competències).** La defensa del treball per part dels alumnes en els lliuraments, el seguiment de l'evolució del treball i la participació activa en els seminaris. S'avaluarà a partir de les entrevistes amb el professor, proves escrites de curta durada, lliuraments o exposicions de problemes i casos plantejats en les activitats. Tot i que aquest instrument és d'avaluació col·lectiva, a criteri del tutor, pot ser corregit individualment en casos en què la participació de l'alumne destaquí (excés o defecte) respecte el seu grup.

Cal assenyalar que mentre els instruments AI i AC1 valoren el rendiment de l'alumne i el grau d'assoliment dels coneixements, l'instrument AC2 valora l'esforç de l'alumne.

INDICADORS I VALORACIÓ:

La qualificació final s'obtindrà segons el següent barem:

$$QF = 30\% AI + 70\% AC$$

$$AC = 60\% AC1 + 40\% AC2 \text{ (valorat per cada lliurament d'activitat)}$$

Per aprovar l'assignatura és necessari haver aconseguit una **puntuació mínima de 5** en les dues qualificacions. A criteri del professor es podrà però establir compensacions entre les notes dels quatre lliuraments parcials de les activitats (corresponents a **AC**).

La data prevista per a l'examen final de l'assignatura servirà per:

- Recuperació de la nota d'avaluació individual AI (examen de teoria).
- Acceptació amb retard dels lliuraments corresponents a les activitats.

CASOS ESPECIALS I RECUPERACIONS:

- Per als instruments d'avaluació col·lectiva (AC) **només hi ha una convocatòria anual**, i està basada principalment en l'avaluació continuada entorn del problema guia i les activitats dels seminaris a lliurar el darrer dia de cada activitat (4 lliuraments). En cas de no acceptació d'un lliurament o retard en els mateixos, els alumnes hauran de re-planificar el seu pla de treball,

disposant com a temps addicional fins al dia de l'examen de febrer de l'assignatura. Aquest retard implicarà penalitzacions en la nota.

- L'avaluació individual consistent en la prova escrita (AI) l'alumne tindrà oportunitat a una segona convocatòria. Per tant, les qualificacions d'AC obtingudes es guardaran fins aquesta segona convocatòria. Atès que la prova escrita es programa per la 6a setmana de curs, la 2a convocatòria es programarà el dia de l'examen de la convocatòria de febrer.
- Atès que l'assignatura se segueix en avaluació contínua, **no es programarà convocatòria de juny**. No es guardarà cap qualificació per al següent curs.

Bibliografia bàsica:

- X. Roca, J. Lladós. **Apunts d'Enginyeria del Software I**. Localitzables a la pàgina web de l'assignatura. Actualitzats Anualment.
- G. Booch, J. Rumbaugh, I. Jacobson. **El lenguaje unificado de modelado**. Addison-Wesley, 1999.
- I. Jacobson, G. Booch, J. Rumbaugh. **El proceso unificado de desarrollo de software**. Addison-Wesley, 1999.
- J. Rumbaugh, I. Jacobson, G. Booch. **El lenguaje unificado de modelado: manual de referencia**. Addison-Wesley, 1999.
- T. Quatrani. **Visual Modeling with Rational Rose 2000 and UML**. Addison-Wesley, 2000.
- P. Krutchen. **The Rational Unified Process. An Introduction**. Addison-Wesley, 2000.
- Roger S. Pressman, **Ingeniería del software, un enfoque práctico**. McGraw-Hill, 4a. edició, 1997.

Bibliografia complementària:

- E. Yourdon, **Análisis Estructurado Moderno**. Prentice-Hall, 1993.
- Lladós, J., Roca, X., **Problemes d'enginyeria del software I**. Servei de Publicacions UAB, 1995.
- Barbee T. Mynatt, **Software engineering with student project guidance**. Prentice-Hall, 1990.
- M.G. Piattini, J.A. Calvo-Manzano, J. Cervera, L. Fernández. **Análisis y diseño detallado de Aplicaciones Informáticas de Gestión**. Ra-Ma, 1996.
- Grady Booch, **Object Oriented Design with applications**. The Benjamin/Cummings Publishing Company, 1990.
- Yourdon Inc., **Yourdon Systems Method Model-driven Systems Development**. Prentice Hall, 1993.
- T. DeMarco, **Structured Analysis and System Specification**. Yourdon Press, 1979.
- I. Sommerville, **Software Engineering**. 3er i 4a Ed., Addison-Wesley, 1992.
- P. Jalote. **An Integrated Approach to Software Engineering**. Springer-Verlag, 1991..
- P. Coad and E. Yourdon, **Object-Oriented Analysis**. Yourdon Press, 1991.
- P. Coad and E. Yourdon, **Object-Oriented Design**. Yourdon Press, 1991.

Enllaços web:

- Pàgina web de l'assignatura: <http://www.cvc.uab.es/shared/teach/a21290/c21290.htm>
- Campus Virtual de la UAB: <https://cv.uab.es/cv/entrada.jsp>