

Enginyeria del Software

2013/2014

Codi: 102743

Crèdits: 6

Titulació	Tipus	Curs	Semestre
2502441 Enginyeria Informàtica	OB	2	2

Professor de contacte

Nom: Josep Lladós Canet

Correu electrònic: Josep.Lladós@uab.cat

Utilització d'idiomes

Llengua vehicular majoritària: català (cat)

Algun grup íntegre en anglès: No

Algun grup íntegre en català: Sí

Algun grup íntegre en espanyol: No

Prerequisits

Atès que l'assignatura és d'introducció al bloc d'assignatures d'enginyeria del software, s'assumirà que no es posseeix cap tipus de coneixement previ sobre la matèria. És responsabilitat de la pròpia assignatura proporcionar als alumnes un mitjà per adquirir els coneixements descrits a l'apartat de continguts de l'assignatura (apartat 6 d'aquesta guia).

Malgrat això és recomanable haver superat les assignatures de Fonaments d'Informàtica i de Metodologia de la Programació de primer curs.

Objectius

Aquesta assignatura és introductòria del bloc d'assignatures sobre el procés d'enginyeria del software. L'objectiu general 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 l'enginyeria del software. S'introduiran les activitats fonamentals que constitueixen el procés de desenvolupament (anàlisi de requeriments, disseny, implementació, proves i manteniment), la majoria de les quals es tracten en més profunditat en assignatures posteriors.

Es tracta per tant que l'alumne adquireixi una visió general de què és un procés de desenvolupament de software, com es modela software, quines eines s'utilitzen, com es gestiona la qualitat i com s'administra un projecte. Més concretament, els objectius són:

- Proporcionar una visió general del procés de desenvolupament del software, els paradigmes, les activitats, i els objectius de cada activitat.
- Conèixer què és el modelat del software. La importància i la dificultat de l'especificació de requeriments i el disseny en les seves diferents vistes.
- Aprendre els conceptes bàsics d'UML per treballar de manera pràctica en el modelat del software.
- Introduir a l'alumne en els principis, conceptes i tècniques que s'apliquen per administrar i controlar la qualitat del software.
- Donar a conèixer les tècniques d'administració requerides per planificar, organitzar, minitoritzar i controlar projectes de software.
- Familiaritzar-se amb els paradigmes de desenvolupament incremental més rellevants: RUP i SCRUM.

Competències

Enginyeria Informàtica

- Analitzar, dissenyar, construir i mantenir aplicacions de manera robusta, segura i eficient, i elegir el

- paradigma i els llenguatges de programació més adequats.
- Capacitat per a comprendre la importància de la negociació, els hàbits de treball efectius, el lideratge i les habilitats de comunicació a tots els entorns de desenvolupament de software.
- Capacitat per concebre, desenvolupar i mantenir sistemes, serveis i aplicacions informàtiques emprant els mètodes de l'enginyeria del software com a instrument per a assegurar-ne la qualitat.
- Capacitat per dissenyar i avaluar interfícies persona-computador que garanteixin l'accessibilitat i usabilitat als sistemes, serveis i aplicacions informàtiques.
- Capacitat per dissenyar, desenvolupar, avaluar i assegurar l'accessibilitat, l'ergonomia, la usabilitat i la seguretat dels sistemes, serveis i aplicacions informàtiques, així com de la informació que gestionen
- Capacitat per resoldre problemes amb iniciativa, presa de decisions, autonomia i creativitat. Capacitat per saber comunicar i transmetre els coneixements, habilitats i destreses de la professió d'enginyer tècnic en informàtica.
- Conèixer i aplicar els principis, metodologies i cicles de vida de l'enginyeria de software
- Treballar en equip

Resultats d'aprenentatge

1. Assumir i respectar el rol dels diversos membres de l'equip, així com els diferents nivells de dependència de l'equip.
2. Comprendre la importància de la negociació, els hàbits de treball efectius, el lideratge i les habilitats de comunicació en tots els entorns de desenvolupament software, a un nivell avançat.
3. Comprendre què és el disseny i com aconseguir un bon disseny que inclogui els requeriments d'accessibilitat, ergonomia, usabilitat i seguretat de les aplicacions informàtiques.
4. Comprendre què és l'anàlisi de requeriments i perquè és important.
5. Conèixer el procés de desenvolupament del software RUP.
6. Conèixer i aplicar mètodes de comunicació i negociació eficaç en la realització de les tasques pròpies de la professió.
7. Conèixer la notació i construcció de diagrames UML.
8. Dissenyar aplicacions sabent seleccionar el paradigma de desenvolupament software més adequat.
9. Dissenyar i avaluar interfases software persona-computador que garanteixin l'accessibilitat i usabilitat dels sistemes, serveis i aplicacions informàtiques.
10. Dissenyar i mantenir aplicacions seguint criteris de robustesa i fiabilitat.
11. Entendre què és el software, quins són els objectius que es plantegen en desenvolupar un software i la problemàtica associada al seu desenvolupament.
12. Treballar cooperativament.

Continguts

BLOC 1. EL SOFTWARE I L'ENGINYERIA DEL SOFTWARE

Tema 1. Principis de l'enginyeria del software.

- 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.

BLOC 2. MODELAT DE SOFTWARE

Tema 2. Anàlisi de Requeriments del Software.

- 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.

Tema 3. Disseny del Software.

- 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.

Tema 4. Introducció a UML (Unified Modeling Language) i RUP (Rational Unified Process)

- Evolució i història d'UML.
- Vistes d'UML. Vista de casos d'ús, vista lògica, vista de components, vista de desplegament.

BLOC 3. ADMINISTRACIÓ DE LA QUALITAT DEL SOFTWARE

Tema 5. Qualitat del software.

- Introducció. Conceptes de qualitat.
- Prova del software. Estratègies de prova, prova d'aplicacions convencionals, prova d'aplicacions web.
- Administració de la configuració del software.

BLOC 4. ADMINISTRACIÓ I DESENVOLUPAMENT DE PROJECTES DE SOFTWARE

Tema 6. Paradigmes de desenvolupament de software.

- Rational Unified Process (RUP). Les 6 best practices. Model estàtic de RUP. Model dinàmic de RUP.
- SCRUM.
- Un cas d'estudi d'UML i RUP.

Tema 7. Administració de projectes del software.

- Conceptes d'administració de projectes.
- Planificació.
- Administració del risc.
- Manteniment i reenginyeria.

Metodologia

L'assignatura d'Enginyeria del Software 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, les diferents activitats i les tasques a dur a terme a cada activitat. 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. No s'aprofundirà en els temes ja que en assignatures de cursos posteriors aquests ja es desenvolupen, però s'explicaran els conceptes necessaris perquè l'alumne entengui el procés de desenvolupament de software complet.

SEMINARIS

Els seminaris són classes amb un nombre reduït d'alumnes. 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.

SEGUIMENT TREBALL PRÀCTIC

No hi haurà sessions presencials de laboratori. Els estudiants resoldran el problema guia de manera autònoma en grups reduïts. Les hores de pràctiques es destinaran a tenir entrevistes individuals setmanals de 30 minuts amb cada equip per fer el seguiment i l'avaluació.

TREBALL PRÀCTIC

Es lliurarà als alumnes a l'inici de curs un problema guia. Aquest problema tindrà dues funcions. En primer lloc utilitzar-lo com a exemple en les classes de teoria i seminaris, resolent-ne parts adients als continguts de cada moment. En segon lloc es pot entendre com l'enunciat d'una pràctica que es va resolent per etapes al llarg del curs. Els alumnes s'organitzaran en equips de treball de 4-5 persones. La resolució del treball la faran de manera autònoma, amb el seguiment setmanal abans descrit.

Activitats formatives

Títol	Hores	ECTS	Resultats d'aprenentatge
Tipus: Dirigides			
Classes de teoria	30	1,2	
Preparació i estudi	30	1,2	
Seminaris / Problemes	15	0,6	
Tipus: Supervisades			
Entrevistes seguiment pràctica	15	0,6	
Tutories	15	0,6	
Tipus: Autònomes			
Treball pràctic	38	1,52	

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), al grau d'implicació en els seminaris i als 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.
- 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:
 - AC1 (avaluació tècnica). Documentació lliurada pels alumnes del seu treball pràctic. S'avaluarà

la resolució tècnica del problema guia.

- 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 destaquï (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:

$$QF = 50\% AI + 50\% AC$$

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

L'assignatura serà avaluada com a No Presentat només en el cas que l'alumne no s'hagi presentat a cap de les proves d'avaluació contemplades ni hagi lliurat totalment o parcialment els treballs.

RECUPERACIONS:

- Avaluació individual (AI). Es farà un examen de teoria alliberatori durant hores lectives. Els alumnes que no superin aquesta prova, disposaran d'un examen de recuperació en la data d'avaluació final programada per la titulació.
- Avaluació col·lectiva (AC). El treball pràctic s'avalua en forma d'avaluació continuada en les sessions de seguiment. Per tant no hi haurà cap activitat de recuperació al final del curs. Tanmateix, el fet de no lliurar a temps o no superar un dels lliuraments intermedis pot ser recuperat reprogramant la planificació del treball de les sessions posteriors durant el curs, d'acord amb el professor.

DATES D'AVALUACIÓ:

Les dates d'avaluació continuada i lliurament de treballs es publicaran al campus virtual i poden estar subjectes a canvis de programació per motius d'adaptació a possibles incidències. Sempre s'informarà al campus virtual sobre aquests canvis ja que s'enten que és el mecanisme habitual d'intercanvi d'informació entre professor i estudiants.

PLAGI:

Sense perjudici d'altres mesures disciplinàries que s'estimin oportunes, i d'acord amb la normativa acadèmica vigent, les irregularitats comeses per un estudiant que puguin conduir a una variació de la qualificació es qualificaran amb un zero (0). Per exemple, plagiar, copiar, deixar copiar, ..., una activitat d'avaluació, implicarà suspendre aquesta activitat d'avaluació amb un zero (0). Les activitats d'avaluació qualificades d'aquesta forma i per aquest procediment no seran recuperables. Si és necessari superar qualsevol d'aquestes activitats d'avaluació per aprovar l'assignatura, aquesta assignatura quedarà suspesa directament, sense oportunitat de recuperar-la en el mateix curs.

Activitats d'avaluació

Títol	Pes	Hores	ECTS	Resultats d'aprenentatge
Avaluació col·lectiva (treball pràctic)	50	4	0,16	1, 2, 4, 5, 6, 7, 8, 10, 12

Avaluació individual (examen teoria)	50	3	0,12	3, 4, 5, 7, 9, 11
--------------------------------------	----	---	------	-------------------

Bibliografia

- 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. Mc Grah-Hill, 7a. edición, 2010.
- A. Álvarez García, R. de las Heras del Dedo, C. Lasa Gómez, Métodos Ágiles y Scrum. Anaya Multimedia, 2012.
- K. Pohl, C. Rupp. Requirements Engineering Fundamentals. Rocky Nook Inc. 2011.