

Gestió del Desenvolupament de Software

Codi: 102789

Crèdits: 3

Titulació	Tipus	Curs	Semestre
2502441 Enginyeria Informàtica	OB	3	2
2502441 Enginyeria Informàtica	OT	4	2

Professor/a de contacte

Nom: Marc Talló Sendra

Correu electrònic: marc.tallo@uab.cat

Utilització d'idiomes a l'assignatura

Llengua vehicular majoritària: català (cat)

Grup íntegre en anglès: No

Grup íntegre en català: Sí

Grup íntegre en espanyol: No

Prerequisits

No hi ha cap prerequisit previ per a cursar aquesta assignatura. Però s'aconsella cursar també les assignatures:

- Arquitectura i tecnologia del software
- Laboratori integrat del software

Objectius

A l'assignatura d'enginyeria del software de segon heu rebut una visió global del procés d'enginyeria del software. Heu après els conceptes bàsics i, per aquells que hagueu cursat les assignatures del primer semestre de la menció d'enginyeria del software, haureu aprofundit en alguns d'aquests conceptes com l'anàlisi de requisits, el disseny i les tècniques de test i control de qualitat.

En aquesta assignatura aprendreu a gestionar el canvi. Tot procés de desenvolupament de software està sotmès a canvis de tot tipus. Canvis provocats per la necessitat de millora del producte desenvolupat. Canvis en els requeriments inicials. Canvis en els terminis o l'equip de treball, pressupost, etc. En definitiva, al final d'aquest curs haureu de ser capaços de gestionar qualsevol canvi que suposi una desviació de la planificació inicial.

Més concretament, els objectius d'aquest curs són:

- Aprofundir en les tècniques de control i desenvolupament del software.
- Aprofundir en el coneixement de les tècniques d'administració necessàries per planificar, monitoritzar i controlar els projectes de software.
- Aprendre a utilitzar algunes de les eines existents per portar a terme el control necessari del desenvolupament de software.

Finalment, pel correcte seguiment d'aquest curs, l'equip docent assumeix que tots els estudiants han assolit les competències treballades a enginyeria del software.

Competències

Enginyeria Informàtica

- Adquirir hàbits de pensament.
- Adquirir hàbits de treball personal.
- Capacitat de solucionar problemes d'integració en funció de les estratègies, estàndards i tecnologies disponibles.
- 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.

Resultats d'aprenentatge

1. Definir i gestionar la documentació que es genera durant el desenvolupament d'una aplicació software.
2. Desenvolupar la capacitat d'anàlisi, síntesi i prospectiva.
3. Desenvolupar un mode de pensament i raonament crítics.
4. Planificar la integració de les diferents components desenvolupades en el procés de codificació.
5. Resoldre problemes d'integració del software amb iniciativa i autonomia.
6. Treballar de manera autònoma.

Continguts

BLOC 1: Gestió de Projectes (GP)

1. Introducció. Conceptes de la GP

Tasques i activitats. Productes, paquets i rols. Model de tasques. Matriu de perfils. Estructures d'organitzacions: tipus i visualització.

2. Activitats de GP

Planificació, organització, control i finalització.

3. Metodologies de GP

Metodologia de Royce. Desenvolupament iteratiu i incremental. Desenvolupament àgil.

BLOC 2: Gestió de la Configuració (GC)

1. Introducció. Conceptes de la GC

Ítems de configuració i agregats de GC. Versions i configuracions. Peticions de canvis. Promocions i *releases*. Dipòsits i espais de treball. Esquemes d'identificació de versions. Eines de GC.

2. Activitats de la GC

Ítems de configuració i identificació dels agregats de la GC. Gestió de la promoció. Gestió dels releases. Gestió de les branques. Gestió de les variants. Gestió del canvi.

3. Gestionant la GC

Documentar la GC. Assignació de responsabilitats en la GC. Organitzar les activitats de la GC.

Metodologia

El procés d'aprenentatge de l'alumne es fonamentarà en aquests dos tipus d'activitats:

Teoria i Problemes. Amb el plantejament de casos d'estudi, els alumnes practican a classe els conceptes teòrics que s'hauran introduït a l'inici de la classe mitjançant suport documental i pissarra. A continuació, els alumnes treballaran en petits grups i discutiran entre ells les possibles solucions fins a concloure i escollir una d'elles. En el tram final de la classe, es posaran en comú totes les solucions plantejades.

Pràctiques. A les classes de pràctiques es treballarà en grups de dos persones i es practicarà l'ús de les eines de gestió de la configuració necessàries en tot projecte de desenvolupament de software.

Tota la documentació i material necessari per al seguiment del curs es trobarà disponible al Campus Virtual.

Competències transversals: Les competències transversals seran treballades i avaluades en diversos moments al llarg del curs. Alguns exemples de com es treballaran són els següents:

- T01.01 - Desenvolupar un mode de pensament i raonament crítics:
A les classes de teoria s'expliquen exemples dels problemes que apareixen a la gestió del desenvolupament de software. S'expliquen les solucions que històricament s'han aplicat i es valoren els pros, contres i errors.
A la pràctica l'alumnat ha d'aconseguir construir un model de desenvolupament del software dintre d'un projecte i ha de ser capaç de trobar una estratègia per gestionar-lo.
A l'examen de teoria hi ha preguntes que obliguen a l'alumne a fer raonaments sobre la matèria per trobar la resposta.
- T02.03 - Gestionar el temps i els recursos disponibles. Treballar de manera organitzada:
Aquesta assignatura tracta sobre com gestionar el temps, recursos i organitzar la feina dintre d'un projecte de desenvolupament de software.
A la segona pràctica l'alumnat ha d'especificar els recursos, la planificació temporal i quina serà la metodologia utilitzada d'un projecte de software.
- T02.04 - Prevenir i solucionar problemes:
Un del temes de l'assignatura tracta sobre com controlar un projecte i prevenir els seus riscos.
A la primera pràctica l'alumnat ha de dissenyar un repositori pensant en els problemes que poden sorgir durant el desenvolupament d'un software concret.
A la segona pràctica l'alumnat ha de gestionar un projecte incloent els possibles canvis de les especificacions que es produiran durant el desenvolupament.
- T02.05 - Prendre decisions pròpies:
A la pràctica, l'alumnat ha de prendre decisions sobre com organitza un repositori de software i com organitza un projecte.

NOTA: La metodologia docent i l'avaluació proposades poden experimentar alguna modificació en funció de les restriccions a la presencialitat que imposin les autoritats sanitàries.

Nota: es reservaran 15 minuts d'una classe, dins del calendari establert pel centre/titulació, per a la complementació per part de l'alumnat de les enquestes d'avaluació de l'actuació del professorat i d'avaluació de l'assignatura/mòdul.

Activitats formatives

Títol	Hores	ECTS	Resultats d'aprenentatge
Tipus: Dirigides			
Classe de Teoria	13	0,52	1, 2, 3, 4, 5
Classe de Problemes	13	0,52	5
Sessions de Pràctiques	6	0,24	1, 2, 3, 4
Tipus: Supervisades			
Tutories	9	0,36	1, 4, 5
Tipus: Autònomes			
Preparació d'exercicis i resolució de supòsits teòrics	15	0,6	3, 5, 6

Avaluació

L'assignatura consta de 2 tipus d'avaluacions: teoria (N_{Teo}) i pràctiques (N_{Lab}). El pes de cadascuna d'aquestes avaluacions és el següent:

$$N_{\text{Final}} = 0.5 \cdot N_{\text{Teo}} + 0.5 \cdot N_{\text{Lab}}$$

$$N_{\text{Lab}} = 0.5 \cdot N_{\text{Lab1}} (\text{SPMP}) + 0.5 \cdot N_{\text{Lab2}}$$

Per calcular la nota N_{Final} cal que la nota de cadascuna de les diferents avaluacions (N_{Teo} i N_{Lab}) siguin totes i cadascuna d'elles igual o superior a 5. En cas contrari, l'assignatura està suspesa i la qualificació final de l'assignatura serà la nota mínima de N_{Teo} i N_{Lab}.

Si N_{Teo} no és superior a 5, l'alumne es podrà tornar a avaluar de N_{Teo} durant les proves de recuperació.

Per a calcular la nota N_{Lab}, totes i cadascuna de les notes N_{Lab}(i), és a dir, totes i cadascuna de les entregues de pràctiques, hauran de ser superiors a 4. En cas contrari, N_{Lab} és la nota mínima de les notes N_{Lab}(i). Les avaluacions mitjançant casos pràctics i la redacció d'informes de pràctiques formen part del bloc d'avaluació de pràctiques (N_{Lab}). Les dates de lliurament i la forma d'entrega dels diferents casos pràctics estaran consignades en els mateixos enunciats. No hi ha segona convocatòria per a la nota N_{Lab} ni per cap de les notes N_{Lab}(i), és a dir, no hi ha segona convocatòria per a les sessions de laboratori.

Per a cada activitat d'avaluació, s'indicarà un lloc, data i hora de revisió en la que l'estudiant podrà revisar l'activitat amb el professor. En aquest context, es podran fer reclamacions sobre la nota de l'activitat, que seran avaluades pel professorat responsable de l'assignatura. Si l'estudiant no es presenta a aquesta revisió, no es revisarà posteriorment aquesta activitat.

Un estudiant es considera No Avaluable únicament si les activitats d'avaluació que ha realitzat estan aprovades (però no ha fet totes les activitats d'avaluació del curs). Si una activitat d'avaluació està suspesa, l'assignatura està suspesa (independentment del nombre d'activitats a les que s'hagi presentat).

La matrícula d'honor es pot aconseguir amb una nota mitjana superior o igual a 9,0. Degut a que hi ha un nombre limitat de matrícules d'honor que es poden donar per grup, s'atorgaran per ordre de nota de major a menor.

Només es convalidarà la nota de pràctiques (N_{Lab}) d'anys anteriors.

Sempre s'informarà al Campus Virtual sobre possibles canvis en el desenvolupament de l'assignatura ja que s'entén que és el mecanisme habitual d'intercanvi d'informació entre professor i estudiants.

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, ..., 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
Examen de Teoria	50%	3	0,12	1, 2, 3, 4, 5, 6
Informes/treballs de pràctiques	50%	1	0,04	1, 2, 3, 4, 5, 6

Bibliografia

- Software war stories : case studies in software management / Donald J. Reifer. Reifer, Donald J., author.; Rutkowski, Michael. 2013 . Biblioteca UAB
- Agile management for software engineering : applying the theory of constraints for business results / David J. Anderson . Anderson, David J. (David James), 1939 cop. 2004 . Biblioteca UAB
- Software configuration management patterns : effective teamwork, practical integration / Stephen P. Berczuk with Brad Appleton . Berczuk, Stephen P.; Appleton, Brad . cop. 2003 . Biblioteca UAB
- Bruegge, B. I Dutoit A.H. - Object-Oriented Software Engineering. Pearson, Prentice-Hall, 2004.
- Berczuk, S. and Appleton B. Software - Configuration Management Patterns: Effective Teamwork, Practical Integration. Addison-Wesley, 2002.
- Steve McConnell - Rapid Development: Taming Wild Software Schedules (Developer Best Practices) - Microsoft Press N° 1 Ed. 1996.

Programari

GIT:

Git és un sistema de control de versions distribuïdes de codi obert i gratuït dissenyat per gestionar des de petits fins a grans projectes amb rapidesa i eficiència.

<https://git-scm.com/>

BITBUCKET:

Bitbucket és més que una simple gestió de codi Git. Bitbucket ofereix als equips un lloc per planificar projectes, col·laborar en el codi, provar i desplegar.

<https://bitbucket.org/product/>