

Enginyeria del software II

Codi	Tipus	Curs/Semestre	Crèdits
20370	Troncal Semestral	3 / 6	5,5

Objectius

Competències específiques

Coneixements

- Aprendre disseny i programació avançada orientats a objecte mitjançant l'estudi de patrons de disseny.
- Aprofundir en els coneixements d'UML mitjançant la utilització d'eines de generació automàtica de codi a partir de dissenys UML.
- Entendre la problemàtica de gestió de la configuració derivada del desenvolupament incremental del software a partir d'un equip de treball. Portar a la pràctica aquesta gestió mitjançant una eina de control de versions.
- Entendre la importància i complexitat de les proves del software. Dissenyar i codificar casos de prova amb eines de suport a la prova automàtica.
- Entendre la importància d'altres temes relacionats amb l'Enginyeria del Software com: interfícies gràfiques d'usuari, disseny per contracte, gestió d'excepcions i estils de codificació.
- Utilització d'eines CASE de suport als diferents temes tractats.

Habilitats

- A partir d'un problema obert i poc especificat, enunciat en llenguatge natural, saber fer un disseny, portar a codi aquest disseny i finalment tenir un executable que respongui a les funcionalitats plantejades al problema. Tot això mitjançant el treball en equip i un entorn integrat de desenvolupament de software.
- Desenvolupar software preparat pel canvi, és a dir, avaluar la qualitat d'un software no només pel seu comportament en execució sinó també per com està fet per dins i el procés de desenvolupament que s'ha seguit.
- Portar a terme totes aquestes activitats en un context de programació orientada a objecte, essent conscient que fer programes en un llenguatge orientat a objecte automàticament no vol dir fer dissenys/programes que realment siguin orientats a objecte.
- Consciència de que un Enginyer en Informàtica no és només un programador i de que fer Software és molt més que programar.
- Saber buscar i seleccionar la informació necessària per assolir els objectius plantejats.

Competències genèriques

- Treball en equip.
- Capacitat d'anàlisi i síntesi.
- Capacitat de resolució de problemes.
- Raonament crític.
-

- Motivació per la qualitat.
- Comunicació oral i escrita.
- Capacitat d'organització i planificació.
- Aprenentatge autònom.

Capacitats prèvies

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

- Programació en llenguatges orientats a objecte (C++, Java).
- Estructures de dades bàsiques.
- Disseny en UML.

Aquests conceptes corresponen a continguts de les assignatures:

- Algorismes i programació
- Llenguatges de programació.
- Estructures de dades.
- Enginyeria del Software I.

Continguts

Introducció als conceptes teòrics	
<ul style="list-style-type: none"> • Patrons de disseny. • Generació automàtica de codi. • Estils de codificació. • Prova. • Gestió de la configuració. • Disseny per Contracte. • Gestió d'excepcions. • Interfícies gràfiques d'usuari (GUIs). • Entorn/Eines de desenvolupament: Eclipse / EclipseUML, JUnit, CVS, Checkstyle, etc. 	

Resolució problema guia	
<ul style="list-style-type: none"> • Estudi individual i en profunditat dels temes teòrics. • Desenvolupament incremental en grup. • Fer dissenys UML amb EclipseUML, generació automàtica de codi amb EclipseUML, programació Java en Eclipse, compliment d'un estil de codificació estàndard (SUN), desenvolupament paral·lel suportat per CVS, codificació de 	

- casos de prova per JUnit i Coverlipse.
- Defensa del treball en reunions i fites.

Nota. En aquest cas "AA" es refereix a "fites + reunions amb el professor".

Metodologia docent

Es vol potenciar la iniciativa i el treball individual i en grup de l'alumne, de manera que aquest sigui la figura principal durant l'aprenentatge i el professor sigui més tutor i facilitador que lector. La idea és que l'alumne vagi adquirint els coneixements i les habilitats de forma continua ajudat pel professor i els seus companys.

Durant tot el semestre l'alumne treballarà en un problema guia amb companys (grup de treball). Així, el procés d'aprenentatge de l'alumne es fonamentarà en aquests dos tipus d'activitats: classes de teoria i resolució de l'esmentat problema guia. La resolució del problema guia s'articula mitjançant el cicle "treball individual/grup => reunió grup i professor => ... => fita d'avaluació => treball individual/grup => reunió grup i professor => ... => fita d'avaluació ...". Queda clar que bàsicament seguim una metodologia tipus PBL (Problem Based Learning).

TEORIA:

Consisteix en classes magistrals amb transparències disponibles al Campus Virtual de la UAB. L'objectiu principal d'aquestes classes és conscienciar a l'alumne dels diferents problemes pràctics associats al desenvolupament professional de software i de les propostes per abordar aquests problemes. Les classes de teoria i la informació que porten associada seran el punt de partida pels alumnes en el moment de cercar la informació destinada a resoldre el problema guia del curs.

REUNIONS DE SEGUIMENT DEL PROBLEMA GUIA:

En aquestes reunions cada grup de treball presenta les activitats fetes per avançar en la resolució del problema guia. El professor avalua la qualitat del treball fet i evita que el grup s'endinsi en una direcció errònia. S'estableix un debat crític entre les propostes dels alumnes i contrapropostes o matisos plantejats pel professor. En tot moment el debat gira al voltant del treball presentat, per tant, el nivell de discussió i de guia varia per cada grup. Podríem dir que el grau de coneixements i comentaris que un grup rep per part del professor és proporcional al treball presentat pel grup (en referència a l'esforç no a la correctesa, és a dir, s'anima a treballar fort encara que ens equivoquem, el més important és poder defensar raonadament el treball presentat).

FITES DE SEGUIMENT DEL PROBLEMA GUIA:

Tenen la mateixa dinàmica que les reunions encara que en aquest cas els alumnes reben una nota de grup pel seu treball.

Avaluació

1a convocatòria (febrer/juny)	2a convocatòria (juliol/setembre)
-------------------------------	-----------------------------------

Avaluació en grups	Avaluació individual	
<ul style="list-style-type: none"> • Hi ha avaluació continuada en grup mitjançant el problema guia ja descrit. • L'avaluació es basa en les reunions setmanals amb el professor i especialment en les fites de lliurament del treball. • Un "No presentat" s'obté si no es fa el problema guia. <p>(més detalls al text principal)</p>	<ul style="list-style-type: none"> • No hi ha avaluació continuada individual, però sí dos exàmens parcials individuals i també una nota que posen els companys de grup. • L'avaluació es basa en les notes tretes als esmentats exàmens i avaluació entre companys. També hi ha casos on es pot anar a un examen de recuperació al Juny. • Un "No presentat" s'obté d'una banda si es falta algun dels exàmens parcials individuals i també a l'examen individual de recuperació; i d'altra banda si es falta a alguna fita o a més de tres reunions. <p>(més detalls al text principal)</p>	<p>No hi ha 2a convocatòria.</p>

Bibliografia bàsica

- Apunts d'Enginyeria del Software II localitzables al campus virtual. Antonio M. López, Joan Serrat. Actualitzats Anualment.
- "Patrones de Diseño". Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Addison-Wesley, 2003.
- "Pragmatic Unit Testing in Java with JUnit". A. Hunt, D. Thomas. The Pragmatic Programmers, 2003.
- "Eclipse in Action". D. Gallardo, E. Brunette, R. McGovern. Manning Publications, 2003.

Bibliografia complementària

- "Head First Design Patterns". Eric Freeman & Elisabeth Freeman. O'Reilly, 2004.
- "JUnit Recipes. Practical Methods for Programmer Testing". J. B. Rainsberger, S. Stirling. Manning Publications, 2004.
- "Design Patterns Explained: a New Perspective on Object-Oriented Design". Alan Shalloway, James R. Trott. Addison-Wesley, 2002.
- "UML y Patrones. Una Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado", 2ª Edición. Graig Larman. Prentice-Hall, 2002.
- "Utilización de UML en Ingeniería del Software con Objetos y Componentes". Perdita Stevens, Rob Pooley. Addison-Wesley, 2002.

- “Thinking in Java”, 3rd edition. B. Eckel. Versió electrònica a www.planetpdf.com, Versió impresa en castellà de Pearson Educación, 2002.
- “The Pragmatic Programmer: from Journeyman to Master”. Andrew Hunt, David Thomas. Addison-Wesley, 2000.

Enllaços
