

Introducció

Grup 40, curs 2010-11 Q2 (P1011)

El professor: Josep Ibarz (a/e: ibarz@lsi.upc.edu i URL: <http://www.lsi.upc.edu/~ibarz>)

La classe: una sessió de teoria (*els dilluns de 19.00 a 21.00 A6202*) i una de laboratori (en el meu cas serà *el grup 42 els dijous de 17.00 a 19.00 A6105 i A6104*). De fet, intentem explicar tota la teoria el més aviat possible, per deixar al final totes les sessions com a laboratori.

Dues recomanacions per a classe: silenci (podeu entrar i sortir quan vulgueu) i diàleg que vol dir em podeu interrompre en qualsevol moment (més val fer una pregunta a temps) us convé vèncer la por al ridícul i les vergonyes inútils.

L'assignatura: el primer que us he de recomanar és una lectura a la guia docent, si encara no li heu fet. Allí hi trobareu la bibliografia recomanada.

La URL de l'assignatura és: <http://www-assig.fib.upc.edu/~prop>. La coordinadora: Alícia Ageno.

És una assignatura pràctica on caldrà *espavilar-se* ja que hi ha coneixements que haureu d'adquirir pel vostre compte i, a més, se us demanarà l'aplicació del vostre *propri criteri*.

Està situada al pla d'estudis després de PRED i en paral·lel a les assignatures d'Enginyeria del *software* (ES). Vol recollir els coneixements adquirits en les assignatures de programació (incloent ADA i BD que es poden fer en paral·lel) per fer una pràctica força més gran (un projecte) i també ser una petita introducció a les ES. [En el pla d'estudis anterior estava entre EDA i ES:E.] Aquells que ja hagueu fet alguna de les assignatures ES veureu que fem un ús força rudimentari dels coneixements que s'hi imparteixen (en aquest cas i per respecte als companys caldrà que us adapteu al nivell de l'assignatura, tot i que particularment en sabeu més).

Som conscients que el nombre de crèdits que té assignats l'assignatura resulten molt justos per la feina que caldrà fer. Per això és important que aprengueu a aprofitar el temps, organitzant-vos el millor possible, cosa que us plantejarà la necessitat d'un bon repartiment de les feines.

També som conscients que si heu seguit estrictament el pla d'estudis, podeu saber poc o gens de gestió de fitxers i bases de dades, però us caldrà fer-ne ús. Per això hi ha força tolerància en aquest aspecte del projecte. (En aquest aspecte, també per respecte, els alumnes amb més coneixements s'hauran d'adaptar al nivell possiblement més baix dels companys del grup.)

Fonamentalment es tracta de fer un projecte, que culmina en un programa que ha d'acabar funcionant sense fallades (ni avortar ni produir resultats erronis).

Es treballa en grup. Per fer el projecte, l'assignatura publica 3 enunciats, cada un d'ells l'ha de fer un grup de 3 alumnes (en algun cas ± 1). Aquests grups de tres alumnes s'agrupen de tres en tres (total 9 alumnes, en el pitjor dels casos ± 3), aquest "supergrup" l'anomenem "cluster". En conjunt, un *cluster* farà els tres enunciats, és a

dir, el primer grup farà el projecte amb el primer enunciat, el segon, amb el segon, i el tercer, amb el tercer.

Tot el *cluster* tindrà el mateix tutor. En les vostres peticions s'ha de respectar el grup de matrícula.

Ala primera sessió de laboratori, un cop presentada la *Visió general del Projecte de Programació* es procedirà a la **formació de grups i clusters**, per això és important d'assistir-hi.

La relació de *clusters* es publicarà al racó de l'assignatura.

Al llarg del curs es fan 3 lliuraments que explicarem a les sessions de laboratori, els dos primers es puntuen globalment en l'àmbit de cada grup, i en el darrer l'alumne individualment. Per això se us exigeix que separeu els fonts que hagi fet cada alumne. En aquest darrer lliurament és l'únic moment en què s'exigeix la presència de tots els membres del grup. La nota final és calcula $10\% 1r + 25\% 2n + 65\% \text{darrer}$.

Només el primer lliurament es pot fer totalment en paper (millor si el feu en un fitxer PDF, en el meu cas ho exigeixo). La resta es fan en disquet (tot i que s'accepti alguna cosa concreta en paper). Cal conèixer i respectar la normativa de l'assignatura que afecta aquests lliuraments.

El tutor pot acceptar o no vostre el treball, en cas de no acceptar-lo us donarà instruccions de com completar-lo perquè pugui ser acceptat. Aquest fet, haver de corregir el lliurament, influirà negativament en la nota. Una primera vegada pot significar que el lliurament es puntuï sobre 6 i una segona, que es puntuï sobre 4.

La assignatura es planteja el projecte com a un projecte real i no com a un simple exercici acadèmic, és a dir, com si en acabar-lo s'hagués de vendre. Per això, a més de les estructures de dades i dels algorismes, es valoren força els aspectes formals com la presentació (aquest aspecte especialment en el primer lliurament), la legibilitat o facilitat de compressió del vostre treball (heu de tenir present que el professor per corregir-lo l'ha d'entendre clarament i amb facilitat), els aspectes ecològics, i d'altres que comentarem i que tenen força pes en un projecte real (habitabilitat o ergonomia de la interfície, documentació, etc.) Tot plegat inclou aspectes que us poden semblar secundaris com la bona redacció dels textos, l'enquadernació o l'ortografia.

Enginyeria: per què diem *enginyeria del software* quan parlem de la construcció de programes i no parlem de fabricació?

En un procés industrial, per exemple per fer una nevera o un cotxe, se sol treballar primer en definir el model i després en fabricar-lo (manufacturar-lo). La primera fase consisteix en definir com ha de ser el producte fins a l'últim detall i en fer un primer exemplar (una nevera o un cotxe que se sotmet a proves de laboratori i reals, i si convé, al carrer). Després cal dissenyar i posar en marxa la cadena de producció, avui en dia en molts casos serà robotitzada, per fabricar el producte que es vol vendre al mercat. A continuació vindrà la producció real de producte i, finalment, caldrà empaquetar-lo i distribuir-lo perquè arribi als clients.

Aquella primera fase en què es defineix el producte i el disseny de la cadena de producció són l'*enginyeria* del producte, un cop posada en marxa la cadena de producció, és quan es fa la *fabricació*. Fixem-nos que quan construïm programes pràctica-

ment no hi ha fabricació, algú podria pensar que la fase de fabricació correspon a la producció del medi de distribució (CD, disquets, FTP, DVD, etc.) i l'execució del procés d'instal·lació, però més aviat això correspondria al fet d'embalar el producte i distribuir-lo. Per tant, diem que a la construcció de *software* només hi ha enginyeria, no hi ha fabricació.

El cicle de vida d'una aplicació informàtica: parlem del cicle de vida d'un producte en definir per quines fases i etapes passa, des de que es comença a pensar o definir fins que es deixa d'usar o deixa de ser útil. En el cas d'una aplicació informàtica inclourà des del moment que es planteja la necessitat o conveniència de fer-la fins aquell en què desinstal·lem o deixem d'usar la darrera còpia.

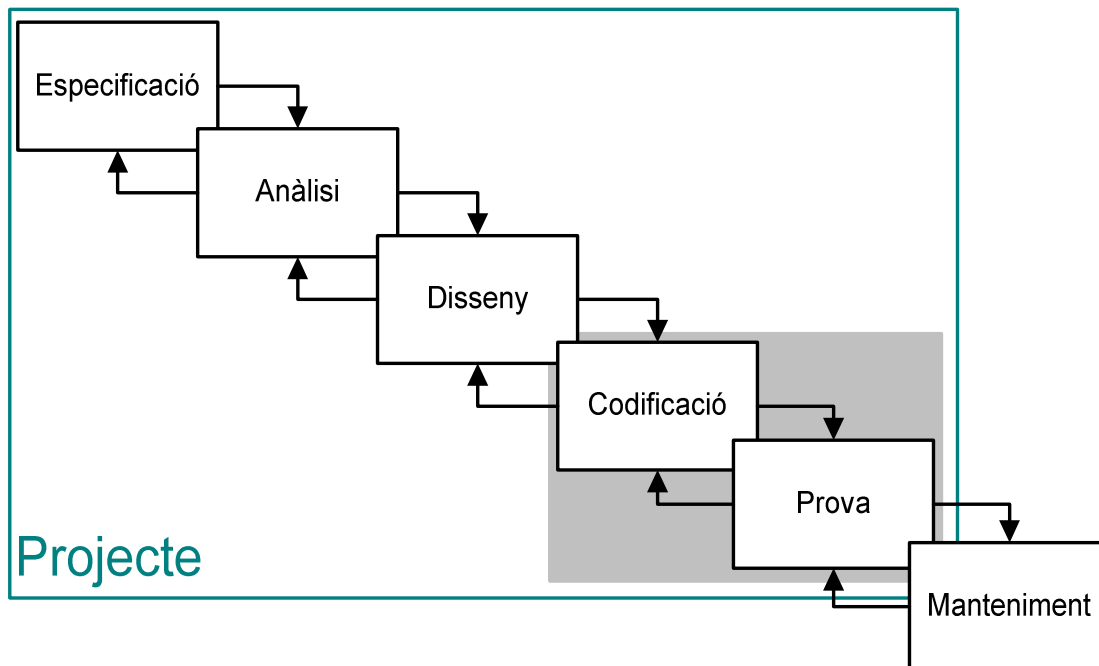
Normalment el cicle de vida forma part i és la guia o l'eix vertebrador de la metodologia de desenvolupament que s'usi. Actualment podem trobar quasi tants cicles de vida com metodologies, però els podríem agrupar tots en un nombre força reduït de famílies. Com que no és objectiu d'aquesta assignatura fer un recorregut pel panorama de les metodologies i dels diferents cicles de vida només parlarem del que s'anomena *cicle de vida clàssic* tot i que actualment es consideri força superat perquè parteix d'idees que avui no són realistes, com l'absència de la possible marxa enrera ja que l'evolució constant de les empreses les força a estar adaptant contínuament les seves aplicacions informàtiques. Tant, que és habitual que abans d'acabar de desenvolupar una aplicació alguns dels requisits o requeriments ja hagin canviat o n'hi hagi de nous.

Malgrat tot, aquest model és vàlid per al curs de projecte de programació ja que el sistema que s'ha d'obtenir té uns requisits prou concrets i és totalment estable.

Aquest cicle de vida també ha rebut el nom de cicle de vida en cascada, perquè es dibuixa en forma d'escala descendent i perquè es considerava inicialment que una etapa començava en acabar l'anterior, no es pensava en una possible marxa enrera, d'una etapa a la següent es passava una documentació molt concreta que era la base de treball però no permetia treballar amb ella o reutilitzar-la, és a dir, les etapes eren disjunctes.

Avui en dia es considera tot el procés com a iteratiu, sempre és possible que des d'una etapa es forci la revisió de l'anterior, per tant, permet la marxa enrera (recorregut ascendent). A més s'ha introduït la idea de transició transparent, que vol dir que la documentació d'una etapa s'usa en la següent com eina directa de treball, es reutilitza i serveix per treballar-hi. Per tant les etapes no són disjunctes, tenen un cert solapament.

L'esquema d'aquest cicle de vida se sol representar així:



1) A l'especificació es defineix el que ha de fer el programa, de fet és una part d'un estudi més ampli que inclourà el cost (econòmic i de temps), l'oportunitat i la definició o especificació del *software*. Aquesta especificació ha d'incloure la definició de la funcionalitat, del que ha de saber fer. Com veurem aquesta etapa és delicada ja que d'ella en depèn l'èxit del projecte, almenys en bona part.

2) L'anàlisi és la primera etapa purament tècnica, en ella s'estudia i defineix a fons la funcionalitat del sistema. En aquesta etapa encara no es tenen en compte els aspectes tecnològics concrets de la implementació (com el llenguatge de programació, S. G. de fitxers o bases de dades, sistema operatiu, etc.) però sí els de funcionament, especialment estructures de dades i tractaments.

3) El disseny és donar-li solució informàtica, per tant definir processos (algorismes), fitxers o bases de dades (amb el S. G. que calgui emprar), etc. Aquí tindran molt de pes les solucions tecnològiques adoptades, el model arquitectònic dels programes, els equips que s'utilitzin, els sistemes operatius, els llenguatges de programació...

4) La codificació és l'escriptura de codi font en els llenguatges de programació que es facin servir.

5) La prova consisteix en comprovar que els programes funcionen correctament, fan el que han de fer (satisfan l'especificació), no avorten mai, etc. De fet, aquesta fase se sol fer conjuntament amb l'anterior, com veurem se sol fer una iteració de codificació i prova.

6) L'etapa de manteniment comença en el moment que es dona el programa per acabat, al final de la posada en marxa, i segueix mentre s'usi el *software*. N'hi ha de quatre tipus:

1.- Correctiu: que vol dir corregir problemes, fallades o defectes d'origen, de construcció.

- 2.- Adaptatiu: adaptar-lo als canvis que es puguin produir (per exemple una nova llei) i que l'afectin.
- 3.- Perfectiu: que vol dir millorar-lo. Més funcionalitat o potenciar la que té.
- 4.- Preventiu: resoldre problemes abans que no es produeixin (com fou el famós efecte 2000).

De fet, es considera que el projecte acaba amb la posada en marxa que és quan comença el manteniment. Convé tenir present que a vegades una operació de manteniment pot desencadenar un nou projecte complet.

Els noms que es donen a les etapes d'aquest cicle de vida no sempre són aquests, podeu trobar-ne d'altres (sinònims), aquests que us presento són els que usem a l'assignatura.

També anirem fent indicacions concretes sobre les adaptacions necessàries per a l'orientació a l'objecte en els moments adequats.

Desenvolupament del projecte: un cop conegut aquest cicle de vida es pot donar un cop d'ull al funcionament de l'assignatura repassant el document "Visió General del projecte de programació" (<http://www-assig.fib.upc.edu/~prop/VisioGeneral.pdf>), cosa que fareu o heu fet a la primera sessió de laboratori.