

Kudeaketaren eta Informazio Sistemen Informatikaren  
Ingeniaritzako Gradua

Bilboko I.I.T.U.E.  
UPV / EHU

**PROGRAMAZIOAREN METODOLOGIA**

**1. maila**

**IG31 taldea**

**2014-15 ikasturtea**

**1. gaia**  
**Sarrera**

Irakaslea: Jose Gaintzarain Ibarmia

Bulegoa: 3I33  
Tutoretza ordutegia: GAUR-en begiratu



# **1. Gaia** **Sarrera**

## **Aurkibidea:**

1.1. Softwarearen garapena .....	5
1.1.1. Helburua .....	5
1.1.2. Softwarearen garapeneko urratsak edo faseak .....	5
1.1.3. Erroreen hedapena .....	5
1.1.4. Mantenimendua .....	6
1.2. Softwarearen zuzentasuna .....	7
1.2.1. Softwarearen zuzentasuna ziurtatzeko teknikak .....	7
1.3. Programazioaren Metodologia .....	7

## **Irudien zerrenda:**

1. irudia: Softwarearen garapenaren faseak .....	6
2. irudia: The problematics of software development .....	8



## **1.1. Softwarearen garapena**

### **1.1.1. Helburua**

Softwarea garatzerakoan problemak ebazten dituzten programa **zuzen** eta eraginkorrak (azkarrak) aurkitzea da helburua.

### **1.1.2. Softwarearen garapeneko urratsak edo faseak**

Softwarearen garapenean 4 fase bereiz daitezke:

- Problemaren analisia.
- Soluzioaren diseinua.
- Soluzioaren implementazioa.
- Mantenimendua.

**Analisi** fasean eraiki behar den sistemaren **espezifikazioa** lortuko da.

Soluzioa **diseinatzeko** fasean **algoritmoa** lortuko da.

**Implementazio** fasean **programa** lortuko da.

**Mantenimendu** fasean aurreko faseetakoren batera itzuli beharko da programan agertutako akatsak konpontzeko edo programari funtzionalitate berriak gehitzeko.

1. irudian (4. orrialdean) ikus daitekeen bezala, softwarea garatzeko prozesuan fase desberdinetatik zehar mugitu beharko da, akatsak eta arazoak agertzen direnean atzera (aurreko faseetara) itzuliz.

### **1.1.3. Erroreen hedapena**

Garapen fase batean burututako **erroreak** ondorengo faseetara hedatzen dira.

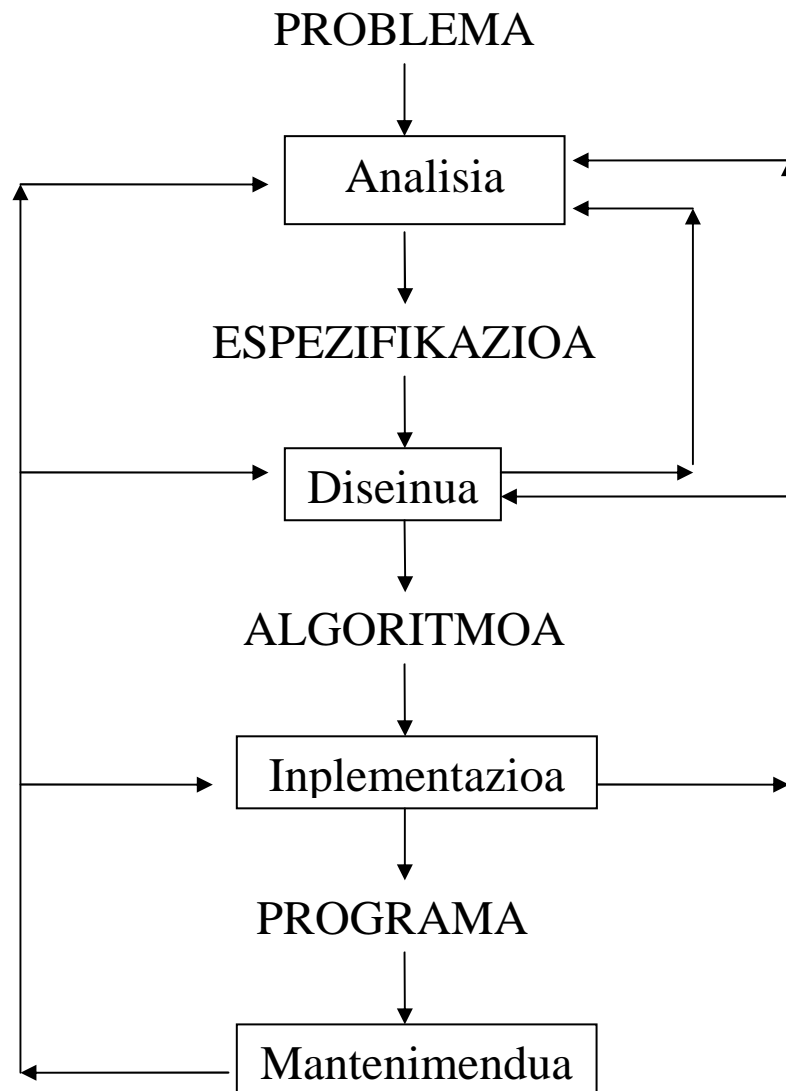
Esate baterako, analisi fasean sistemaren espezifikazio okerra lortzen bada, espezifikazio hori kontuan hartuz diseinu fasean eraikitako algoritmoa ez da egokia izango hasierako problema ebazteko edo benetan nahi den sistema eraikitzeko.

Era berean algoritmo okerra jarraituz implementatutako programa ez da egokia izango hasierako problema ebazteko.

Hasierako faseetan egindako akatsak edo hanka sartzeak izaten dira garestienak. Problema gaizki ulertu bada eta lortutako espezifikazioa ez bada egokia, hasierako problema ebazteko egokia ez den programa eraikiko da azkenean. Garatutako programa egokia ez dela konturatutakoan, analisi fasera itzuli beharko da, erroreak konpondu eta berriro fase denak zeharkatuz joateko.

Fase batean erroreren bat edo ondo ez doan zerbait aurkitutakoan atzera joan beharko da erroreak iturburua aurkitu arte.

Adibidez, inplementazio garaian aurkitutako arazo bat diseinu fasean algoritmo okerra eraiki izanaren ondorioa izan daiteke baina gerta daiteke arazoa hasieran problema gaizki ulertzeagatik eta ondorioz espezifikazio okerra lortzeagatik izatea ere.



**1. irudia: Softwarearen garapenaren faseak**

#### **1.1.4. Mantenimendua**

Programa bat garatu ondoren, lehenago edo geroago aldaketa eta hobekuntzen beharra sortzen da askotan. Horren arrazoia programari funtzionalitate berriak gehitu nahia edo lehendik dituen funtzionalitateak eguneratu nahia izan ohi da.

Programa bati funtzionalitate berriak gehitzeko edo lehendik dituenak aldatzeko programak zer egiten duen oso ondo ulertu behar da. Programa egiterakoan hartu ziren erabakiak eta erabaki horien zergatiak ezagutzea komeniko da.

Gainera oso ohikoa da softwarearen mantenimenduaz arduratuko direnak software hori garatu zutenak ez izatea.

Hori guztia dela eta oso garrantzitsua da softwarearen garapen prozesu dena oso era zehatzean dokumentatzea, horrela software hori aldatu beharko dutenei gauzak asko erraztuko baizaizkie.

<b>Programak dokumentatzearen beharra</b>
---

## **1.2. Softwarearen zuzentasuna**

Programa batek egin beharko lukeena eta programa horrek benetan egiten duena bat datozenean programa hori zuzena dela esaten da.

### ***1.2.1. Softwarearen zuzentasuna ziurtatzeko teknikak***

Softwarearen garapen prozesuan zuzentasuna ziurtatzen lagunduko diguten tresnak edukitzeko helburuarekin, honako teknika hauek landuko ditugu:

- **Programen espezifikazioa**: programa batek zer egin behar duen zehazteko.
- **Programen dokumentazioa**: programen puntu desberdinetan zein propietate betetzen diren adierazteko.
- **Programen egiaztapena**: programa bat zuzena al den ala ez erabakitzeke.
- **Programen eratorpena**: programa zuzenak eta programen zuzentasun frogak aldi berean garatzeko.
- **Datu-moten espezifikazioa**: zein datu erabiliko diren, datu horiekin zein eragiketa buru daitezkeen eta datu horiek eragiketa horiekiko zein propietate betetzen dituzten adierazteko.
- **Programen eraldaketa**: programa zuzen bat hartu abiapuntu bezala eta programa hori eraldatuz programa berria lortzeko. Programa berriak hasierakoren gauza bera egingo du baina azkarragoa izango da edo beste ezaugarri interesanteren bat izango du.

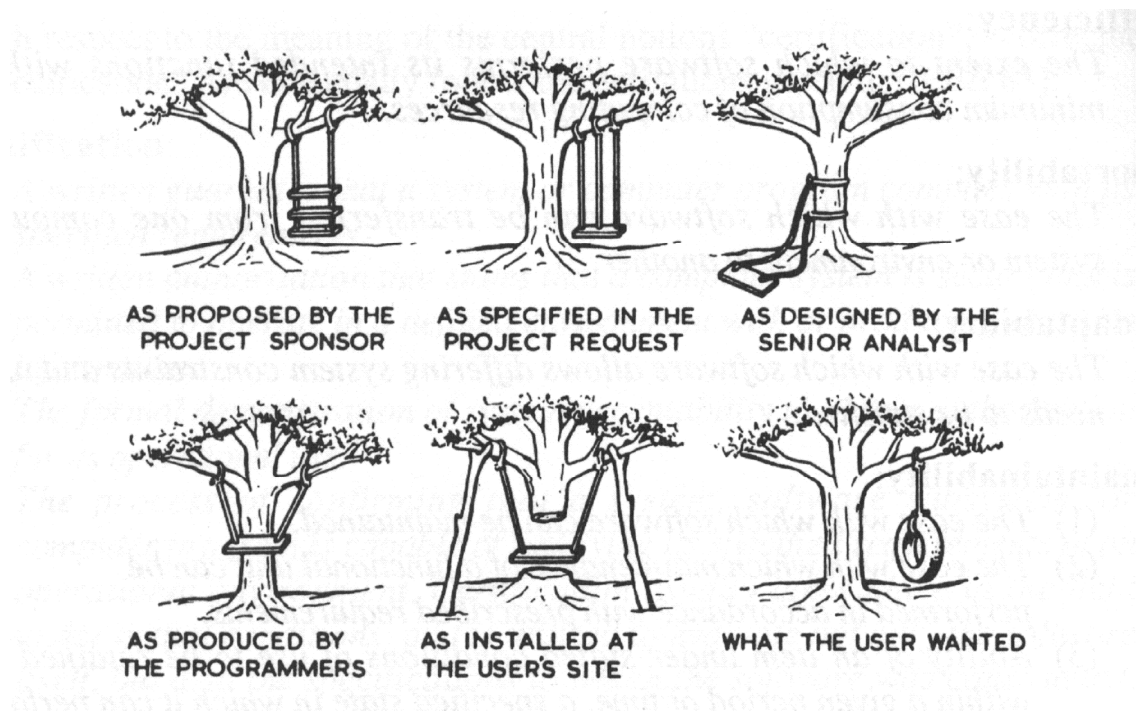
## **1.3. Programazioaren Metodologia**

Programazioaren metodologiak PROGRAMA ZUZENAK era metodiko, arrazonatu eta eraginkorrean espezifikatu, diseinatu, analizatu eta dokumentatzeko balio duten teknikak, kontzeptuak eta metododoak biltzen ditu.

Irakasgaien landutako kontzeptu eta metodo garrantzitsuenak honako hauek dira:

- Programen **aurre-ondoetako espezifikazioa eta dokumentazioa**: **Lehen Mailako Logikaren** lengoaia erabiliz (2. gaia)
- Programen **egiaztapena**: **Hoare-ren kalkulua** erabiliz (3. gaia)
- Programen **eratorpen formalak**: **Hoare-ren kalkulua** erabiliz (4. gaia)
- Datu-moten **espezifikazio ekuazionala**: **Haskell** programazio-lengoaia funtzionala erabiliz (5. gaia)
- Programa **errekurtsiboak iteratibo bihurtzea**: **Burstall-en metodoa** jarraituz. (6. gaia)

\*\*\*\*\*



## 2. irudia: The problematics of software development

"Specification and Transformation of Programs"  
Helmut A. Partsch, Springer-Verlag, 1990