AGRICULTURAL MANAGEMENT SYSTEM
PROJECT ESTIMATION

Project size estimation methods :

- LOC-based estimation
- FP-based estimation
- Empirical estimation models
- COCOMO model

LOC-based Estimation:

The problems of lines of code (LOC)

- Different languages lead to different lengths of code
- It is not clear how to count lines of code
- A report, screen, or GUI generator can generate thousands of lines of code in minutes

LOC-based Estimation for Online Agricultural Management System:

Total number of Line in Project= 20000

FUNCTION	ESTIMATED LOC
User interface	1800
2D geometric analysis	2500
3D geometric analysis	3100
Database management	3280
Graphic display facilities	3600
I/O control function	2900
Analysis function	2820
Total estimated LOC	20000

FP based Estimation(Function Point Analysis):

The basic and primary purpose of the functional point analysis is to measure and provide the software application functional size to the client, customer, and the stakeholder on their request. Further, it is used to measure the software project development along with its maintenance, consistently throughout the project irrespective of the tools and the technologies.

The Five Components of Function Points

- **Data Functions**
  - Internal Logical Files(ILF)
  - External Interface Files(ELF)
- **Transactional Functions**
  - External Inputs(EI)
  - External Outputs(EO)
  - External Inquiries(EQ)

Function Point Analysis of Online Agricultural Management System:

- Unadjusted Function Point Contribution:

TRANSFER FUNCTIONS	FIELD TYPE/FILE INVOLVED	FTRs	DETs	COMPLEXITY	UFP
Search(EQ)	Fields:Searchtext,Item name,Price,Type,Picture,Submit File: Item	1	6	Low	3
Display(EQ)	Fields: Item name,owner info,Price,Product described name,Product review,Buynow,Rating File: Item	1	7	Low	3
Add to cart(EI)	Fields:Itemname,Type,Price, Picture, Quantity Files: Cart,Item	2	5	Average	4
Buynow(EI)	Fields:Name,city,Mobileno,pincode, Email,Address Files: Order,cart	2	6	Average	4
Viewblogs(EQ)	Fields: Blogtext,Like,Submit,Comment	1	4	Low	3

Writeblogs(EI)	Fields: Blogtext,Submit	1	2	Low	3
Add/Modify (2*EI)	Fields: ItemName,Description,Price,Quantity, Submit,Confirm,Picture File: Item	1	7	2*Low	6
Our product(EQ)	Fields: Itemname,Type,Price,Picture	1	4	Low	3
				TOTAL	29

❖ Unadjusted Function Point Contribution

DATA FUNCTIONS	**FIELD TYPE/FILE INVOLVED**	**FTRs**	**DETs**	**COMPLEXITY**	**UFP**
Item(ILF)	Fields:Itemname,Price,Description, Picture,Quality	1	5	Low	7
Cart(ILF)	Fields:Itemname,Price,Picture,Type, Quantity	1	5	Low	7
Order(ILF)	Fields:Text,Home	1	2	Low	7
				TOTAL	21

❖ Performance and Environmental impact

GSC	**DI**
Data communication	2
Distributed Data Processing	2
Performance	2
Heavily used configuration	2
Transaction rate	2
Online Data Entry	2
End user Efficency	2
Online update	2
Complex processing	2
Reusability	2
Installation Ease	2
Operational Ease	2
Multiple Sites	2
Facilitate change	2
Total	**28**

VALUE ADJUSTMENT FACTOR(VAF)=(0.65+(0.01*Total))
=(0.65+(0.01*28))
= 0.93

UFP =UFP(Data Fn)+UFP(Transaction fn)
=21+29
=50

Adjusted Function Point Count = UFP* VAF
= 50*0.93
= 46.5

Effort = AFP * Productivity
= 46.5*10.6
= 492.9 per hours

An Empirical Estimation Model:

The structure of empirical estimation models is a formula, derived from data collected from past software projects, that uses software size to estimate effort. Size, itself, is an estimate, described as either lines of code (LOC) or function points (FP). No estimation model is appropriate for all development environments, development processes, or application types. Models must be customised (values in the formula must be altered) so that results from the model agree with the data from the particular environment.

The typical formula of estimation models is:

$$E = a + b(S)^c$$

where;

E represents effort, in person months,
S is the size of the software development, in LOC or FP, and
a, b, and c are values derived from data.

AN EMPIRICAL ESTIMATION MODEL FOR ONLINE AGRICULTURAL MANAGEMENT SYSTEM:

a= 2.4; b= 1.05; c= 2.5;S=20

E= $2.4 + 1.05(20)^{2.5} = 1880.69$ person months.

COCOMO MODEL

CO- Constructive

CO-Cost

MO-Model (Constructive Cost Model)

It is based on LOC(Lines of Code) where project estimation is done based on the total lines of codes required to develop the system i.e. Size of the system define the cost of the project.

It is used to estimate the **effort,cost,development time,average staff size,productivity** etc.

There are 3 types of Cocomo model

1.BASIC COCOMO MODEL

2.INTERMEDIATE COCOMO MODEL

3.DETAILED COCOMO MODEL

BASIC COCOMO MODEL

It estimates the software roughly and quickly.It is mostly useful for small-medium sized software.

There are 3 modes of development

1)Organic -Project size(2-50 KLOC)

2)Semi-detached- Project size(50-300 KLOC)

3)Embeded- Project size(Over 300 KLOC)

Effort = a(KLOC)^{b} Person-Month

Development Time = c(Effort)^{d} Months

Average Staff Size = Effort / Development time Persons

Productivity = KLOC / Effort KLOC/person-month

Basic COCOMO Co- efficients

Project	a_b	b_b	c_b	d_b
Organic mode	2.4	1.05	2.5	0.38
Semidetached mode	3.0	1.12	2.5	0.35
Embedded mode	3.6	1.20	2.5	0.32

BASIC COCOMO MODEL ESTIMATION FOR ONLINE AGRICULTURAL MANAGEMENT SYSTEM:

KLOC=20

Effort = a(KLOC)^b Person-Month

Effort = $2.4(20)^{1.05}$

=2.4 * 23.2317

=55 Person/Month(appx)

Development Time = c(Effort)^d Months

Development Time = $2.5(55)^{0.38}$

=11.46

=11 Persons(appx)

Average Staff Size = Effort / Development time Persons

Average Staff Size = 55/11 Persons

= 5 Persons.

Productivity = KLOC / Effort KLOC/person-month

=20/55

=0.363(appx)

INTERMEDIATE COCOMO MODEL

The intermediate model estimates software development effort in terms of size of the program and other related cost drivers parameters (product parameter, hardware parameter, resource parameter, and project parameter) of the project. The estimated effort and scheduled time are given by the relationship:

Effort (E) = $a*(KLOC)^b$ *EAF MM

Scheduled Time (D) = $c*(E)^d$ Months(M)

Where,

- **E =** Total effort required for the project in Man-Months (MM).
- **D =** Total time required for project development in Months (M).
- **KLOC =** The size of the code for the project in Kilo lines of code.
- **a, b, c, d =** The constant parameters for the software project.
- **EAF =** it is an effort adjustment factor which is calculated by multiplying the parameter value of different cost driver parameters

Cost Drivers	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	1.07	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Project attributes						
Application of software engineering methods	1.24	1.10	1.00	0.91	0.82	
Use of software tools	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

INTERMEDIATE COCOMO MODEL ESTIMATION FOR ONLINE AGRICULTURAL MANAGEMENT SYSTEM:

Given the estimated size of the project is =20 KLOC
Having high application experience =0.82
Having very low experience in programming=1.14
EAF=0.82*1.14=0.9348
Effort (E) = $3.0*(20)^{1.12}$ * 0.9348 MM
=80.3514MM

Scheduled Time(D)= $2.5*(80.3514)^{0.35}$ Months(M) =11.60 Months(M)