18CS2009 – 20CS2050L Software Engineering Lab URK20CS2001

Ex. No: 8	**ESTIMATION OF PROJECT SIZE USING FUNCTION POINT ANALYSIS & COCOMO MODEL**
Date: 27/09/2021	
Video Link:	https://drive.google.com/file/d/1TmRDZgnvQTdGU1oxYWlq5R8TtY2eXoVz/view?usp=sharing

OBJECTIVE

The objective is to calculate the estimation of project size using function analysis & COCOMO mode.

This document will clearly explain the Function point analysis and COCOMO model for the project E-Learning platform.

METHODOLOGY

A project would be fall in one of the scenarios.

1. Much relevant project data is available for the current project but not much information about previous projects.
2. Previous project data ae available for the project but not much information about the current project.
3. Project data are available for the current project as well as that of the previous projects.
4. Some project data are available for the current projects.
5. No project data are available for both current as well as previous projects.

Estimation Technique Selection Based on Project Information Availability:

	Project Details	**Estimation Technique**
01	Historical project data & current project data	Function Point Analysis
02	Current project data	COCOMO, Wide Band, Delphi
03	No data	No Technique

18CS2009 – 20CS2050L Software Engineering Lab URK20CS2001

DESCRIPTION

(i)Function point analysis

The function point is a "unit of measurement" to express the amount of business functionality an information system (as a product) provides to a user. Function points are used to compute a functional size measurement (FSM) of a software. The cost (in dollars or hours) of a single unit is calculated from past projects.

Step-1: Calculate F where

Scale varies from 0 to 5 according to character of Complexity Adjustment Factor (CAF). Below is the scale:

0 – No Influence
1 – Incidental
2 – Moderate
3 – Average
4 – Significant
5 – Essential

Step – 2: Calculate Complexity Adjustment Factor (CAF):

Step – 3: Calculate Unadjusted Function Point (UFP) by multiplying each individual function point to corresponding values in the table.

Measurement Parameter	**Weighting factor**		
	Simple	**Average**	**Complex**
No. of user inputs	3	4	6
No. of user outputs	4	5	7
No. of user inquiries	3	4	6
No. of files	7	10	15
No. of external interfaces	5	7	10

Step – 4: Calculate Function Point (FP)

FP = UFP * CAF

Upon Calculating FP, it is used to calculate productivity and cost.

(ii) Basic COCOMO model

The COnstructive COst Estimation MOdel (COCOMO) model gets the number of estimated lines of code for the project and calculates the overall time and people required for project. The

output differs based on the different project type and the different types of projects are as follows.

(a) Organic: Relatively small, simple software projects in which small teams with good application experience work to a set of less than rigid requirements.

(b) Semi-detached: An intermediate, (in size and complexity), software project in which teams with mixed experience levels must meet a mix of rigid and less than rigid requirements.

(c) Embedded: A software project that must be developed within a set of tight hardware, software and operation constraints

The COCOMO models calculates the no of person (effort) and duration using the equations 6 &

7.

where

• KLOC is the estimated size of the software product expressed in Kilo Lines of Code

• a, b, c, d are constants for each category of software products and their values are as follows.

Project type	a	b	c	d

Organic	2.4	1.05	2.5	0.38
Semi-detached	3	1.12	2.5	0.35
Embedded	3.6	1.2	2.5	0.32

• Tdev is the estimated time to develop the software, expressed in months

• Effort is the total effort required to develop the software product, expressed in person months (PMs)

OUTPUT (Manual Calculation):

Step 1 & 2: CAF Calculation

		Weightages
1.	Does the system require reliable backup and recovery?	**2**
2.	Are data communications required?	**3**
3.	Are there distributed processing functions?	**0**
4.	Is performance critical?	**0**
5.	Will the system run in an existing, heavily utilized operational environment?	**0**
6.	Does the system require on-line data entry?	**5**
7.	Does the on-line data entry require the input transaction to be built over multiple screens or operations?	**5**
8.	Are the master files updated on-line?	**5**
9.	Are the inputs, outputs, files, or inquiries complex?	**3**
10.	Is the internal processing complex?	**3**
11.	Is the code to be designed reusable?	**5**
12.	Are the conversion and installation included in the design?	**4**
13.	Is the system designed for multiple installations as different organizations?	**0**
14.	Is the application designed to facilitate change and ease of use by the user?	**0**
	Total(F)	**35**

CAF = 0.65 + (0.01 * F)

CAF = 0.65 + (0.01 * 35)

CAF = 1

Step 3: UFP Calculation

Measurement Parameter	Count	Weighing Factor (WF)	Count x WF
No. of User Input	10	6	60
No. of user output	10	7	70
No. of user inquires	25	6	150
No. of files	50	15	750
No. of external interfaces	3	10	30
		UFP	1060

Step 4: FP, Productivity & Cost calculation.

FP = UFP * CAF = 1060 * 1

FP = 1060

OUTPUT (Verification):

Domain Characteristic Table

MEASUREMENT PARAMETER	COUNT (value >= 0)	WEIGHTING FACTOR Simple	Average	Complex
Number of User Input	10	○	○	◉
Number of User Outputs	10	○	○	◉
Number of User Inquiries	25	○	○	◉
Number of Files	50	○	○	◉
Number of External Interfaces	3	○	○	◉

Complexity Adjustment Table | FP Calculation

Complexity Adjustment Table

ITEM	COMPLEXITY ADJUSTMENT QUESTIONS	SCALE No Influence 0	1	2	3	4	Essential 5
1	Does the system require reliable backup and recovery?	○	○	◉	○	○	○
2	Are data communications required?	○	○	○	◉	○	○
3	Are there distributed processing functions?	◉	○	○	○	○	○
4	Is performance critical?	◉	○	○	○	○	○
5	Will the system run in an existing, heavily utilized operational environment?	◉	○	○	○	○	○
6	Does the system require on-line data entry?	○	○	○	○	○	◉
7	Does the on-line data entry require the input transaction to be built over multiple screens or operations?	○	○	○	○	○	◉
8	Are the master files updated on-line?	○	○	○	○	○	◉
9	Are the inputs, outputs, files or inquiries complex?	○	○	○	◉	○	○
10	Is the internal processing complex?	○	○	○	◉	○	○
11	Is the code to be designed reusable?	○	○	○	○	○	◉
12	Are conversion and installation included in the design?	○	○	○	○	◉	○
13	Is the system designed for multiple installations in different organizations?	◉	○	○	○	○	○
14	Is the application designed to facilitate change and ease of use by the user?	◉	○	○	○	○	○

Domain Characteristic Table | FP Calculation

FP Calculation

NOTE: For any updates made on any of the entries, always click the 'Calculate Function Points' button to recalculate function points value.

Reset / Clear all form entries Calculate Function Points

RESULT	
PROJECT FUNCTION POINTS	1060

Top of Page | Domain Characteristic Table | Complexity Adjustment Table

Output (Calculation-Organic mode):

KLOC = 5

Effort = a*KLOC^{b} = 2.4 * $5^{1.05}$ = 2.4 * 5.42

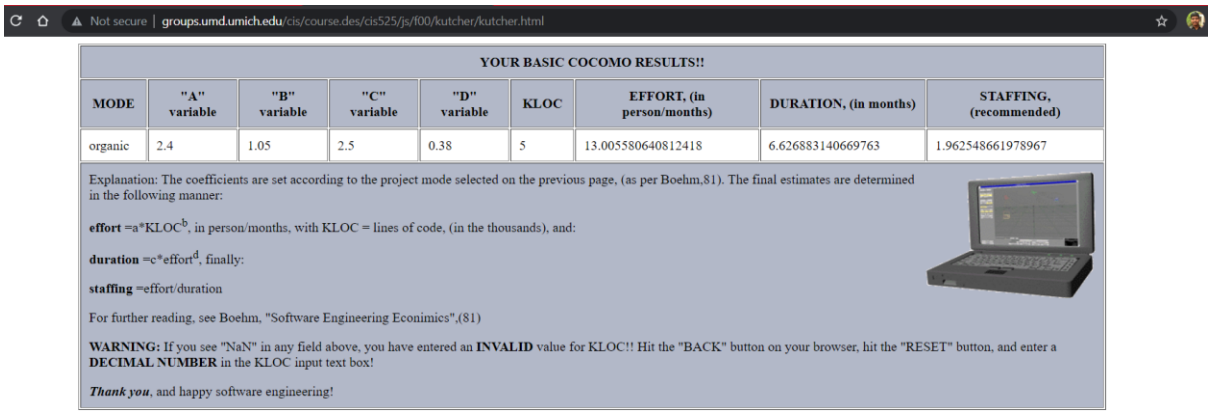
Effort = 13.01

Duration = c * effort^{d} = 2.5 * $13.01^{0.38}$ = 2.5 * 2.65

Duration = 6.63 months

Staffing = effort/duration = 13.01/6.63

Staffing = 1.96 persons



Output (Calculation Semi-detached mode):

KLOC = 5

Effort = a * KLOC^{b} = 3 * $5^{1.12}$ = 3 * 6.07

Effort = 18.21

Duration = c*Effort^{d} = 2.5 * $18.21^{0.35}$ = 2.5 * 2.76

Duration = 6.9 months

Staffing = Effort/duration = 18.21/6.9

Staffing = 2.64 persons

Not secure | groups.umd.umich.edu/cis/course.des/cis525/js/f00/kutcher/kutcher.html

YOUR BASIC COCOMO RESULTS!!								
MODE	**"A" variable**	**"B" variable**	**"C" variable**	**"D" variable**	**KLOC**	**EFFORT, (in person/months)**	**DURATION, (in months)**	**STAFFING, (recommended)**
semi-detached	3	1.12	2.5	0.35	5	18.19565356758946	6.90126476845646	2.636567959362485

Explanation: The coefficients are set according to the project mode selected on the previous page, (as per Boehm,81). The final estimates are determined in the following manner:

effort =a*KLOC^b, in person/months, with KLOC = lines of code, (in the thousands), and:

duration =c*effort^d, finally:

staffing =effort/duration

For further reading, see Boehm, "Software Engineering Econimics",(81)

WARNING: If you see "NaN" in any field above, you have entered an **INVALID** value for KLOC!! Hit the "BACK" button on your browser, hit the "RESET" button, and enter a **DECIMAL NUMBER** in the KLOC input text box!

Thank you, and happy software engineering!

Output (Calculation Embedded mode):

KLOC = 5

Effort = a*KLOC^b =3.6 * $5^{1.2}$ = 3.6 * 6.9

Effort = 24.84

Duration = c*effort^d = 2.5 * $24.84^{0.32}$ = 2.5 * 2.8

Duration = 7 months

Staffing = Effort/duration = 24.84/7

Staffing = 3.55 persons

Not secure | groups.umd.umich.edu/cis/course.des/cis525/js/f00/kutcher/kutcher.html

YOUR BASIC COCOMO RESULTS!!								
MODE	**"A" variable**	**"B" variable**	**"C" variable**	**"D" variable**	**KLOC**	**EFFORT, (in person/months)**	**DURATION, (in months)**	**STAFFING, (recommended)**
embedded	3.6	1.2	2.5	0.32	5	24.835133906301866	6.98813506903734	3.5539000979445956

Explanation: The coefficients are set according to the project mode selected on the previous page, (as per Boehm,81). The final estimates are determined in the following manner:

effort =a*KLOC^b, in person/months, with KLOC = lines of code, (in the thousands), and:

duration =c*effort^d, finally:

staffing =effort/duration

For further reading, see Boehm, "Software Engineering Econimics",(81)

WARNING: If you see "NaN" in any field above, you have entered an **INVALID** value for KLOC!! Hit the "BACK" button on your browser, hit the "RESET" button, and enter a **DECIMAL NUMBER** in the KLOC input text box!

Thank you, and happy software engineering!

According to the Domain characteristic table, Complexity adjustment table, and the COCOMO model, This E-Learning system has achieved the Functional point result of 1060.

And by using the Cocomo model this E-Learning platform achieved the organic values, semi detached values and the Embedded values successfully and got the required results.

RESULT:

The estimation of project was performed using function point analysis and COCOMO model.