Ex.No: 08

Estimation of project size using Function Point Analysis & Cocomo Model

CS2050 L Software Engineering Lab

Why project size estimation??

Project details

A project would be fall in one of the scenarios

1. Much relevant project data are available for the current project but not much information about previous projects.

2. Previous project data are 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 previous projects.

4. Some project data are available for the current project.

5. No project data are available for both current as well as previous projects.

Summary

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

Project Size Estimation

1. FP Calculator
2. COCOMO

Project Size Estimation

1. FP Calculator
2. COCOMO

Function Point

- Function points were defined in 1979 in Measuring Application Development Productivity by Allan Albrecht at IBM.
- 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.

http://groups.umd.umich.edu/cis/course.des/cis525/js/f00/harvey/FP_Calc.html

Software Engineering
Tiny Tools

FP CALCULATOR

Domain Characteristic Table

MEASUREMENT PARAMETER	COUNT (value >= 0)	WEIGHTING FACTOR Simple	Average	Complex
Number of User Input	12	○	○	◉
Number of User Outputs	60	○	○	◉
Number of User Inquiries	9	○	○	◉
Number of Files	0	○	○	◉
Number of External Interfaces	0	○	○	◉

Complexity Adjustment Table | FP Calculation

Software Engineering

Tiny Tools

FP CALCULATOR

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

Software Engineering

Tiny Tools

FP CALCULATOR

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?	◉	○	○	○	○	○

Software Engineering

Tiny Tools

FP CALCULATOR

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?	◉ ○ ○ ○ ○ ○

Software Engineering

Tiny Tools

FP CALCULATOR

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

Software Engineering
Tiny Tools

FP CALCULATOR

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	354.90000000000003

Top of Page | Domain Characteristic Table | Complexity Adjustment Table

Step-1:

F = 14 * scale

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

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

Step-2:

Calculate Complexity Adjustment Factor (CAF).

$CAF = 0.65 + (0.01 * F)$

- **Step-3: Calculate Unadjusted Function Point (UFP)**

TABLE 1: Function point complexity weights.

Measurement parameter	Weighting factor		
	Simple	Average	Complex
Number of user inputs	3	4	6
Number of user outputs	4	5	7
Number of user inquiries	3	4	6
Number of files	7	10	15
Number of external interfaces	5	7	10

- **Multiply each individual function point to corresponding values in TABLE.**

Step-4: Calculate Function Point.

- FP = UFP * CAF
- FP = UFP * (0.65 + (0.01 * F))

Project Size Estimation

1. FP Calculator
2. COCOMO

Project Size Estimation

1. FP Calculator
2. COCOMO

COCOMO

- Boehm proposed COCOMO (Constructive Cost Estimation Model) in 1981.
- COCOMO is one of the most generally used software estimation models in the world.
- COCOMO predicts the efforts and schedule of a software product based on the size of the software.

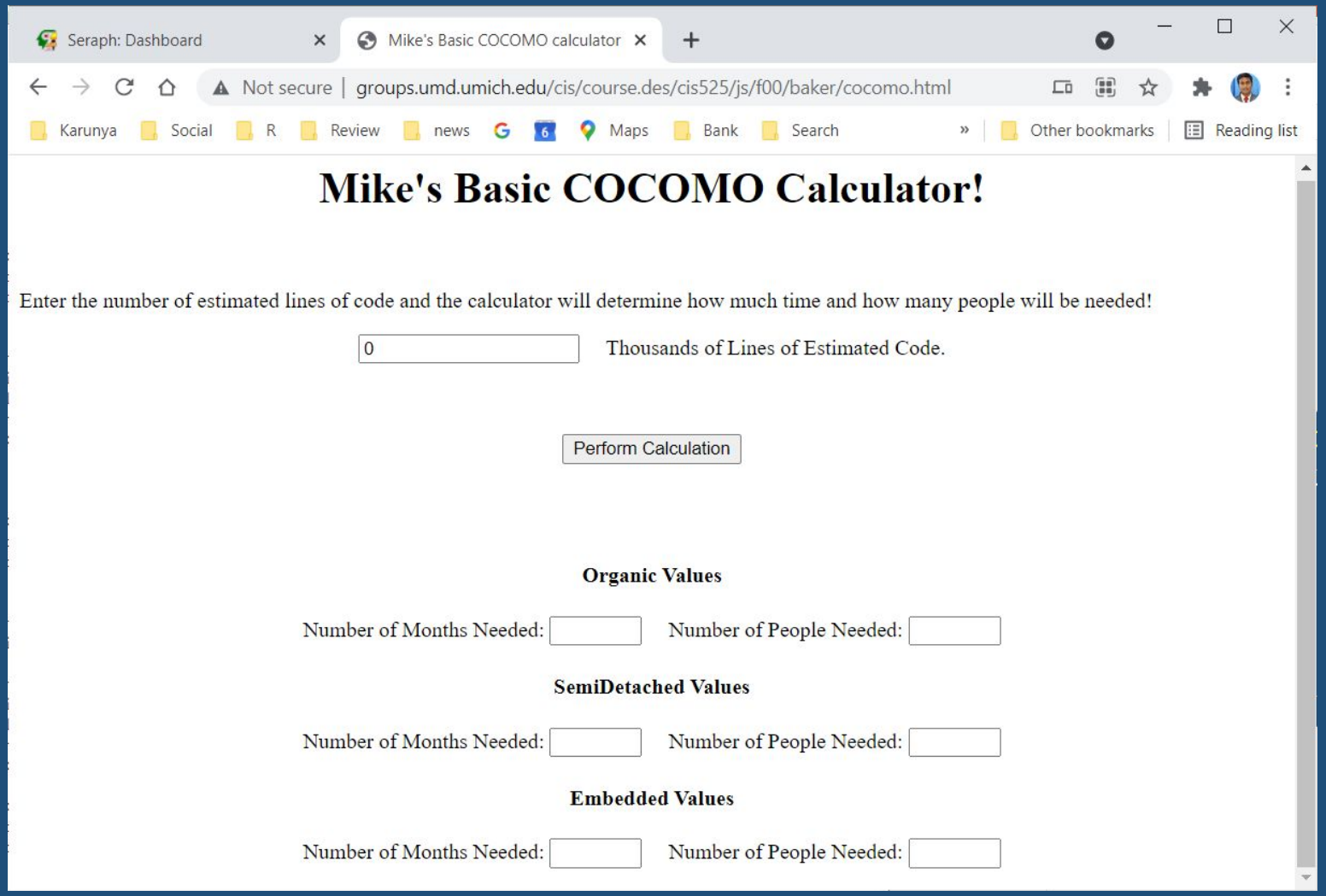
(http://groups.umd.umich.edu/cis/course.des/cis525/js/f00/baker/cocomo.html)

COCOMO

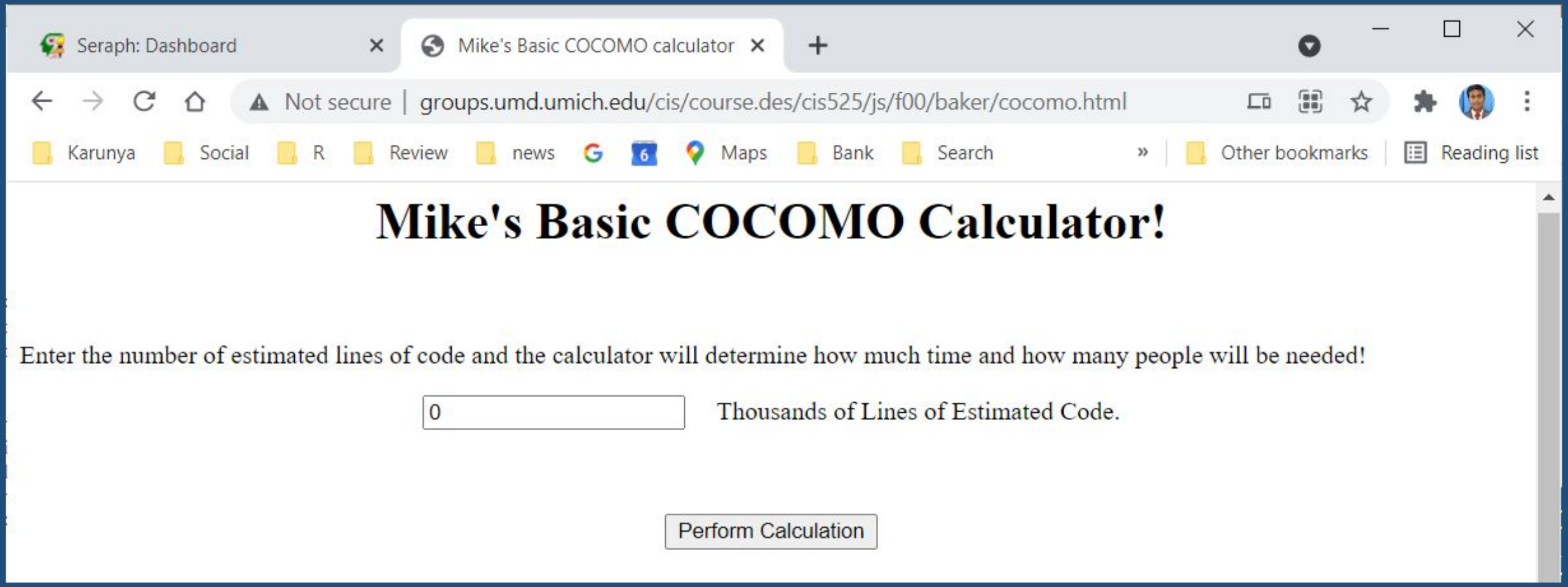
- Boehm proposed COCOMO (Constructive Cost Estimation Model) in 1981.
- COCOMO is one of the most generally used software estimation models in the world.
- COCOMO predicts the efforts and schedule of a software product based on the size of the software.

(http://groups.umd.umich.edu/cis/course.des/cis525/js/f00/baker/cocomo.html)

COCOMO


Seraph: Dashboard
Mike's Basic COCOMO calculator
Not secure | groups.umd.umich.edu/cis/course.des/cis525/js/f00/baker/cocomo.html
Karunya
Social
R
Review
news
Maps
Bank
Search
Other bookmarks
Reading list
Mike's Basic COCOMO Calculator!
Enter the number of estimated lines of code and the calculator will determine how much time and how many people will be needed!
0
Thousands of Lines of Estimated Code.
Perform Calculation
Organic Values
Number of Months Needed:
Number of People Needed:
SemiDetached Values
Number of Months Needed:
Number of People Needed:
Embedded Values
Number of Months Needed:
Number of People Needed:

COCOMO


Seraph: Dashboard
Mike's Basic COCOMO calculator
Not secure | groups.umd.umich.edu/cis/course.des/cis525/js/f00/baker/cocomo.html
Karunya
Social
R
Review
news
Maps
Bank
Search
Other bookmarks
Reading list
Mike's Basic COCOMO Calculator!
Enter the number of estimated lines of code and the calculator will determine how much time and how many people will be needed!
0
Thousands of Lines of Estimated Code.
Perform Calculation

COCOMO

Organic Values

Number of Months Needed: [ ] Number of People Needed: [ ]

SemiDetached Values

Number of Months Needed: [ ] Number of People Needed: [ ]

Embedded Values

Number of Months Needed: [ ] Number of People Needed: [ ]

Different Modes of Output

- **Organic:** Relatively small, simple software projects in which small teams with good application experience work to a set of less than rigid requirements.
- **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.
- **Embedded:** A software project that must be developed within a set of tight hardware, software and operation constraints.

COCOMO - Calculation

Effort (E) = $a*(KLOC)^b$ 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 a software project.

PROJECT TYPE	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semidetached	3	1.12	2.5	0.35
Embedded	3.6	1.2	2.5	0.32