Name: Taimoor Khan

ID: 13579

BSSE 7th sem

Software Project Management

Submitted to: Sir Zain Shaukat

Ans 1: Cost-benefit Analysis:

It is a standard way to assess the economic benefits

It consists of two steps:

- Identify and estimate all the costs and benefits of carrying out the project
- Express the costs and benefits in a common unit for easy comparison

Costs

- Development costs
- Setup costs
- Operational costs

Benefits

- Direct benefits
- Assessable indirect benefits
- Intangible benefits

Cost-benefit Evaluation Techniques Example:

Year	*Project 1*	*Project 2*	*Project 3*	*Project 4*
0	-100,000	-1,000,000	-100,000	-120,000
1	10,000	200,000	30,000	30,000
2	10,000	200,000	30,000	30,000
3	20,000	200,000	30,000	30,000

4	20,000	200,000	20,000	25,000
5	100,000	350,000	20,000	50,000
Net Profit	60,000	150,000	30,000	45,000
Payback	5	5	4	4
ROI	12%	4%	6%	7.5%

Ans 2: Function Point Analysis (FPA)

FPA was Developed by A. Albrecht in IBM. The main aim is to estimate the LOC of a system (line of code). LOC of system= FP of system × LOC-per-FP of the language

FPA is a top-down approach.

Developed by Albrecht (1979) and later refined by Albrecht and Gaffney (1983)

Used in development with 3GL (3rd Generation Language)

LOC means Line of Code (programming statement)

For COBOL, the LOC per FP is 91.

For C, the LOC per FP is 128.

Idea: Software system consists of five major components (or, *external user types*)

- External input types
- External output types

- Logical internal file types
- External interface file types
- External inquiry types

- Identify each instance of each external user type in the proposed system
- Classify each instance as having high, medium or low complexity
- Assign the FP of each instance
- FP of the system = sum of FP of individual components

Function Point Analysis

Number of FPs	Complexity		
External user type	Low	Average	High
External input type	3	4	6
External output type	4	5	7
Logical internal file type	7	10	15
External interface file type	5	7	10
External inquiry type	3	4	6

Function Point Analysis – Example

- A component of an inventory system consisting of ‘Add a record’, ‘Delete a record’, ‘Display a record’, ‘Edit a record’, and ‘Print a record’ will have
    - 3 external input types
    - 1 external output type
    - 1 external inquiry type

    Then, assign FPs based on the complexity of each type

Object Point Analysis

- Similar to function point analysis
- Used on 4GL development projects
- Takes account of features that may be more readily identifiable if the system is built on high-level application building tools

Steps:

- Identify the number of screens, reports and 3GL components
- Classify each object as Simple, Medium and Difficult
- Assign the weight accordingly
- Calculate the total object points

Total OP = sum of individual OP × weighting

- Deduct the reused objects (r% reused)

NOP = OP × (1 – r%)

- Identify the productivity rate of both developer and CASE
- Productivity rate = average of the two PRs
- Calculate the effort

Effort = NOP / Productivity Rate

	Number and source of data tables		
Number of views contained	Total < 4 (<2 server, <2 client)	Total < 8 (2-3 server, 3-5 client)	Total 8+ (>3 server, >5 client)
< 3	Simple	Simple	Medium
3 – 7	Simple	Medium	Difficult
8+	Medium	Difficult	Difficult

Object Point Analysis – Screens

Object Point Analysis – Reports

	Number and source of data tables

Number of sections contained	Total < 4 (<2 server, <2 client)	Total < 8 (2-3 server, 3-5 client)	Total 8+ (>3 server, >5 client)
< 2	Simple	Simple	Medium
2 or 3	Simple	Medium	Difficult
> 3	Medium	Difficult	Difficult

Object Point Analysis – Complexity Weightings

	Complexity		
Type of object	Simple	Medium	Difficult
Screen	1	2	3
Report	2	5	8
3GL component	N/A	N/A	10

Object Point Analysis – Productivity Rate

	Very low	Low	Nominal	High	Very High
Developer's experience and capability	4	7	13	25	50
CASE maturity and capability	4	7	13	25	50

Ans3:

Cocomo (Constructive cost model):

It is a parametric cost model in which important aspects of software projects are characterized by variables (or parameters). Once the value of the parameters are determined, the cost can be computed from an equation.

It recognizes different approaches to software development such as

Prototyping, Incremental development etc.

History:

COCOMO originally proposed by Boehm in 1981, now called COCOMO 81

Later evolved to *Ada COCOMO* in 1989

In 1995, Boehm proposed COCOMO II

- The basic model equation

Effort = Constant × $(\text{Size})^{\text{scale factor}}$ × Effort Multiplier

  - Effort in terms of person-months
  - Constant: 2.45 in 1998
  - Size: Estimated Size in KSLOC
  - Scale Factor: combined process factors
  - Effort Multiplier (EM): combined effort factors

Example:

Suppose that a project was estimated to be 400 KLOC. Calculate the effort and development time for each of the three modes that is

1.Organic

2.Semidetached

3.Embedded

Solution: The basic COCOMO equation is

$E= a_b(KLOC)^{bb}$

$D= c_b(E)^{db}$

Estimated size of the project = 400 KLOC

1.Organic Mode: $E=2.4(400)^{1.05} = 1295.31$ PM

$D=2.5(1295.31)^{0.38} = 38.07$ M

2.Semi detached Mode: $E=3.0(400)^{1.12} = 2462.79$ PM

$D=2.5(2462.79)^{0.35} = 38.54$ M

3.Embedded Mode: $E=3.6(400)^{1.20} = 4772.81$ PM

$D=2.5(4772.81)^{0.32} = 37.59$ M

Advantages	Disadvantages •
Good improvement over COCOMO Good match for iterative development, modern technology, and management process	Still immature, diverse projects in database Hard to believe that it will be any more reliable than the original COCOMO model