Name : Haider Ali
ID : 6916
Department : computer science
Subject : SPM

Question no 1: Explain the cost benefit analysis with an example solved.

Ans: Cost benefit Analysis is a standard way to assess the economic benefits. It is a technique used to compare the total costs of a programme with its benefits, using a common matric. This analysis includes two major steps.

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

Example:

year	Project 1	Project 2	Project 3
0	-200,000	-500,000	-1000,000
1	50,000	100,000	200,000
2	50,000	100,000	200,000
3	40,000	140,000	300,000
4	40,000	200,000	150,000
5	50,000	100,000	250,000
Net profit	30,000	140,000	100,000
Payback	5	4	5
ROI	4%	6%	5%

Question no 2: What is FPA and OPA? Provide examples

Ans: Functional Point Analysis is one of the mostly preferred and widely used estimation technique used in the software engineering. FPA is used to make estimate of the software project, including its testing in the terms of functionality or function size of the software product.

It consists of five major components.

i. External input types
ii. External output types
iii. Logical internal file types
iv. External interface file types
v. External inquiry types

Example:

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 file type	3	4	6

OPA:

i. Similar to function point analysis
ii. Used on 4GL development projects
iii. 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

Example:

Screen and Report Classification

Object point complexity levels for screens

	Number and sources of data tables		
Number of Views Contained	**Total < 4**	**Total < 8**	**Total 8+**
<3	simple	simple	medium
3-7	simple	medium	difficult
8+	medium	difficult	difficult

Object point complexity levels for reports

	Number and source of data tables		
Number of Sections Contained	**Total < 4**	**Total < 8**	**Total 8+**
0-1	simple	simple	medium
2-3	simple	medium	difficult
4+	medium	difficult	difficult

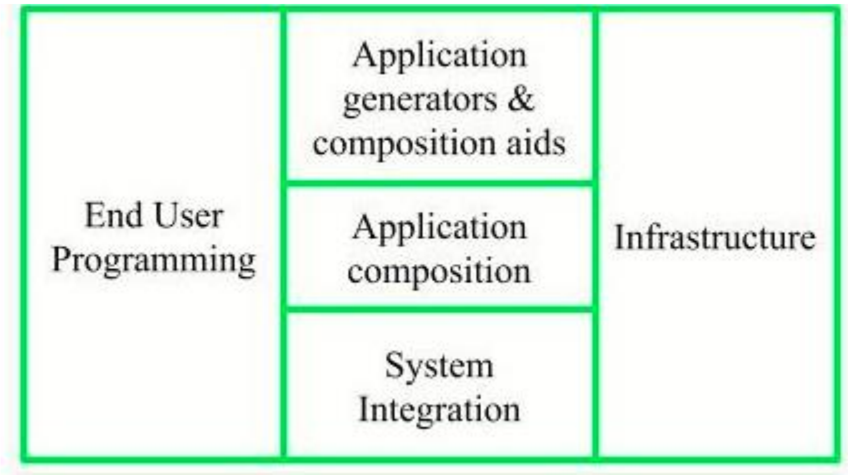
Object Points Computation

	Weight		
Type	**Simple**	**Medium**	**Difficult**
Screen	1	2	3
Report	2	5	7
Modules	10	10	10

Question no 3: COCOMO II

COCOMO-II is the revised version of the original Cocomo (Constructive Cost Model) and is developed at University of Southern California. It is the model that allows one to estimate the cost, effort and schedule when planning a new software development activity.

It consists of following components



End user programming:

Application generators are used in this sub-model. End user write the code by using these application generators.

Example – Spreadsheets, report generator, etc.

Intermediate sector:

(a). Application Generators and Composition Aids –

This category will create largely prepackaged capabilities for user programming. Their product will have many reusable components. Typical firms operating in this sector are Microsoft, Lotus,
Oracle, IBM, Borland, Novell.

(b). Application Composition Sector –

This category is too diversified and to be handled by prepackaged solutions. It includes GUI, Databases, domain specific components such as financial, medical or industrial process control packages.

(c). System Integration –

This category deals with large scale and highly embedded systems.

Infrastructure Sector:

This category provides infrastructure for the software development like Operating System, Database Management System, User Interface Management System, Networking System, etc.

Stages of COCOMO II:

1. Stage-I:

   It supports estimation of prototyping. For this it uses Application Composition Estimation Model. This model is used for the prototyping stage of application generator and system integration.

2. Stage-II:

   It supports estimation in the early design stage of the project, when we less know about it. For this it uses Early Design Estimation Model. This model is used in early design stage of application generators, infrastructure, system integration.

3. Stage-III:

   It supports estimation in the post architecture stage of a project. For this it uses Post Architecture Estimation Model. This model is used after the completion of the detailed architecture of application generator, infrastructure, system integration