Lecture 05

Evaluating the Proposal

- Once the proposals are generated, they are evaluated

- It is necessary to check
  1. Technical Feasibility
  2. Operational Feasibility
  3. Economical Feasibility

- Risk Analysis

- Identify the areas (uncertain) of high risks for which uncertainty about getting successful outcome
  - Identify and Special attention take

Economical Feasibility

- **Cost Benefit Analysis**

- Cost : Place actual money value against any purchase or activity needed to implement the project

- Benefit : Place money value against any benefits that will accure from a new system created by the project

- Two steps
  1. Estimation of Costs and Benefits
  2. Determine whether a project is worthwhile or not

Estimation of Costs and Benefits

- Goal is to produce a list of what is required to implement the project
- List of the new system benefits
- Evaluating Costs
  - Tangible Items (Direct value can be attached)
  - Intangible Items (Direct value can’t be attached , guess)
- Evaluating Benefits
  - Tangible Items (Direct value can be attached)
  - Intangible Items (Direct value can’t be attached , guess)

Cost of Tangible Items

1. Equipment costs for the new system
   - Computing equipments costs, accommodation costs, furniture costs etc.

2. Personnel Costs – Analyst, Designers, Programmers etc.

3. Material Costs – Stationary, manuals, documents etc.

4. Conversion Cost – New forms, Procedures etc.

5. Training Costs – Users, developers etc.

6. Other Costs – Consultants costs, travel, management overhead etc.

- Value can’t be precisely determined & subjective judgment
  - How much save to complete the project earlier
  - Provide new information to the decision makers

Benefits of Tangible & Intangible Items

1. Measure by actual value
   - Reduce of production cost
   - Reduce the processing cost

2. Can’t be measure (only subjective judgement)
   - Increase the sales in the market
   - Wider market of marketing data
   - Maintain good business image

Determine whether a project is worthwhile or not

1. Payback method (Forward)

2. Present Value method(backward)

- **Payback method (Forward)**
  - Define time required to recover the initial cost
  - We know how much a project will cost to start
  - Know cost and benefits for each succeeding year

- **Example**
  - Cost at start is $50,000 (project tenure is 5 years)
  - Benefits - $10,000 at the end of 1st year
    - $20,000 at the end of 2nd year
    - $30,000 at the end of 3rd year
    - $10,000 at the end of 4th year
    - $5,000 at the end of 5th year

payback period is between 2nd and 3rd year

Determine whether a project is worthwhile or not (Cont.)

- We assume for banking system, and rate of interest is 10% per annum
- **Present Value method(backward)**
- Determine how much money it is worthwhile for investing now in order to receive a given return in some years time
- Project benefits are estimated for each year from today
- Then compute the present value of savings
- If the present cost exceeds the present value, then it is not worthwhile

Determine whether a project is worthwhile or not (Cont.)

- Example
  - At t=0, Initial investment is $50,000
  - At t=1, Benefit = $10,000 then the amount deposited in the 0th year= $10000/(1+10/100)
  - At t=2, Benefit = $20,000 then the amount deposited in the 0th year= $20000/ $(1+10/100)^2$
  - At t=3, Benefit = $30,000 then the amount deposited in the 0th year= $30000/$(1+10/100)^3$
  - At t=4, Benefit = $10,000 then the amount deposited in the 0th year= $10000/$(1+10/100)^4$
  - At t=5, Benefit = $5,000 then the amount deposited in the 0th year= $,5000/ $(1+10/100)^5$
- Sum of the present value > Initial investment

COCOMO (COnstructive COst MOdel) Model

- COCOMO I
- Barry Boehm introduce it
- Hierarchy of software estimation model
- Model takes the following form :
  1. Model1(Basic COCOMO) : is a static single valued model that computes software development effort (and cost) as a function of program size expressed in estimated line of code
  2. Model2(Intermediate COCOMO) :computes software development effort as a function of program size and a set of "cost drivers" that include subjective assessments of product, hardware, personnel and project attributes

COCOMO (COnstructive COst MOdel) Model

- Model3(Advanced COCOMO) :incorporates all characteristics of the intermediate version with an assessment of the cost driver’s impact on each step (analysis, design etc.) of the software engineering process

- COCOMO may applied to 3 classes of the software projects :

1. Organic mode: Small, simple s/w project
   - Ex. Tharmal analysis program
2. Semi-detached mode: Intermediate size and complexity
   - Ex. Transaction processing
3. Embedded mode:

COCOMO (COnstructive COst MOdel) Model

- Basic COCOMO equation take the form :
- Effort= $E = a_b (KLOC) \exp(b_b)$ person-months
- Duration= $C_b (E) \exp(d_b)$ months
- Cost driver attributes:

1. Product attributes

   I. Required s/w reliability
   II. Size of application database
   III. Complexity of the product

COCOMO (COnstructive COst MOdel) Model

2. H/W attributes
   - I. Run-time performance constraints
   - II. Memory constraints
   - III. Volatility of the virtual m/c environment
   - IV. Required turnaround time
3. Personnel attributes
   - I. Analyst capability
   - II. S/w engineer capability
   - III. Applications experience
   - IV. Virtual m/c experience
   - V. Programming Language experience
4. Product attributes
   - I. Use of s/w tools
   - II. Application of S/w engineering methods
   - III. Required development schedule

Attributes value (Table by Boehm)

S/w project	a_a	b_b	c_c	d_d
Organic	2.4	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Attributes value (Table by Boehm)

S/w project	a_i	b_i
Organic	3.2	1.05
Semidetached	3.0	1.12
Embedded	2.8	1.20

COCOMO (Cont.)

- Each of the 15 attributes is rated on a 6-point scale that ranges from **very low** to **extra high**
- Based on the rating **effort multiplier** is determined from table published by Boehm
- Product of effort multipliers results is an effort adjustment factor (EAF) ranges from 0.9 to 1.4
- Intermediate COCOMO Equation takes the form :

$$E = a_i (KLOC) \exp(b_i) \times EAF$$

COCOMO II

- Hierarchy of estimation models that address the following areas :

  1. **Application composition model** : used during early stage of s/w engg., when prototyping of user interfaces, consideration of s/w and system interaction, assessment of performance and evaluation of technology maturity are paramount.
  2. **Early design stage model** : used once requirements have been stabilized and basic s/w architecture has been established
  3. **Post- architecture stage model** : used during the construction of the s/w

COCOMO II (Cont.)

- Sizing information :
  1. Object points
  2. Function points
  3. Lines of source code

- The COCOMO II in application composition modes uses object points

- An indirect s/w measure that is computed using counts of the number of (1) **Screens** (at the user interface), (2) **Reports** and (3) **Components** (likely to be required to build the application)

COCOMO II (Cont.)

- Each object instance (screen or report) is classified into one of three complexity levels

Object type	Complexity weight		
	Simple	Medium	Difficult
Screen	1	2	3
Report	2	5	8
Components			10

- **Object point count** is determine by multiplying the original number of object instances by the weighting factor in the table

- Summing to obtain total object point

- When component based development or general s/w reuse is to be applied, the % of reuse is estimated and the object point count is adjusted

COCOMO II (Cont.)

- New Object Point, NOP=(Object Point) X [(100-%reuse)/100]
- Productivity Rate, PROD=NOP/Person-month
- Estimated Effort, E= NOP/PROD
- Productivity rate :

Developer’s Experience/Capacity	Very low	Low	Nominal	High	Very High
Environment Maturity/Capacity	Very low	Low	Nominal	High	Very High
PROD	4	7	13	25	50

Problems

1. Using COCOMO I
   - Determine the Effort to develop of a Software product
   - Determine the Duration to develop of a Software product
   - Determine the Number of People engaged to develop of a Software product.

- Input of your program is Lines of Code and Effort Adjustment Factor.

- Also determine the type of project (i) Organic, (ii) Semi-detached and (iii) Embedded.

Problems

- Using COCOMO II
  - Determine the Object Point to develop of a Software product
  - Determine the New Object Point to develop of a Software product
  - Determine the Effort to develop of a Software product
  - Determine the Number of People engaged to develop Software product, if Duration of development of the software is 5 years.
- If the software consist of 10 Screens, 4 Reports and 15 3GL Components. Assume component based development and 60% reuse is applied.