SOFTWARE ENGINEERING

Software Project Management

Objective

Enable group of software developers to work efficiently towards successful completion of the project.

Job of Project Manager

- Project Planning
- Project Monitoring

Project Planning

- Estimation
    1. Cost
    2. Duration
    3. Effort
- Scheduling
- Staffing
- Risk management
- Miscellaneous plan

Measurement problems

- Estimating the size of the measure
- Estimating the total number of programmer months which have elapsed
- Estimating contractor productivity (e.g. documentation team) and incorporating this estimate in overall estimate

Lines of code

- What's a line of code?
  - *The measure was first proposed when programs were typed on cards with one line per card*
  - *How does this correspond to statements as in Java which can span several lines or where there can be several statements on one line*
- What programs should be counted as part of the system?
- Assumes linear relationship between system size and volume of documentation

Function points

- Based on a combination of program characteristics
  - *external inputs and outputs*
  - *user interactions*
  - *external interfaces*
  - *files used by the system*
- A weight is associated with each of these
- The function point count is computed by multiplying each raw count by the weight and summing all values

Function Point Computation

- UFP (Unadjusted Function Point)

UFP = (No. of inputs) *4 + (No. of outputs) * 5 + (No. of inquiries) * 4 + (No. of Files) *10 + (No. of interfaces) *10

- TCF (Technical Complexity Factor)

14 factors are there which can have influence index in the range 0 to 6.

Total degree of influence is calculated as DI (Degree of Influence)

TCF = 0.65 +0.01*DI 0<=DI<=84

0.65<=TCF <=1.35

- FP (Function Point)

FP= UFP * TCF

Project Estimation Techniques

1. Empirical Estimation Technique

   Expert Judgement Technique,

   Delphi

2. Heuristic Technique

   COCOMO

3. Analytical Estimation Technique

   Halstead’s Software estimation

Expert judgement

- One or more experts in both software development and the application domain use their experience to predict software costs. Process iterates until some consensus is reached.
- Advantages: Relatively cheap estimation method. Can be accurate if experts have direct experience of similar systems
- Disadvantages: Very inaccurate if there are no experts!

COCOMO

Different models of COCOMO have been proposed to predict the cost estimation at different levels, based on the amount of accuracy and correctness required. All of these models can be applied to a variety of projects, whose characteristics determine the value of constant to be used in subsequent calculations.

Boehm’s Classification

- Organic
- Semidetached
- Embedded

COCOMO - I

- Model E = a S^b * m(x)

	BASIC		INTERMEDIATE	
MODE	a	b	a	b
Organic	2.4	1.05	3.2	1.05
Semidetached	3.0	1.12	3.0	1.12
Embedded	3.6	1.20	2.8	1.20

Basic COCOMO

- Computes software development effort (and cost) as a function of program size, expressed in estimated lines of code.
- m(x) = 1

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.
- $m(x) = \pi\ m(x_i)$

Cost Drivers	Rating					
	Very Low	Low	Nominal	High	Very High	Extra High
Product Attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Database size		0.94	1.00	1.08	1.16	
Product complexity	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes						
Execution time constraint			1.00	1.11	1.30	1.66
Main storage constraint			1.00	1.06	1.21	1.56
Virtual machine volatility*		0.87	1.00	1.15	1.30	
Computer turnaround time		0.87	1.00	1.07	1.15	
Personnel Attributes						
Analyst capabilities	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Programmer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience*	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Project Attributes						
Use of modern programming practices	1.24	1.10	1.00	0.91	0.82	
Use of software tools	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

Detailed COCOMO

- Includes all characteristics of the intermediate version with an assessment of the cost driver's impact on each step (analysis, design, ect.) of the software engineering process
- m(x) based on similar questionnaire

Halstead’s Software Science

Assumptions

- complete algorithm specification exists
- programmer works alone
- programmer knows what to do
- Based on

  N = # of unique operators

  n = # of unique operands

Halstead Equations

Effort

$E = N^2 * log_2 (n) / 4$

To compute N

$N = k * S_s$

k = average # operators per LOC

k is language specific

To compute n

$N = n * log_2 (n / 2)$

To be continued..