UNIT-3

Metrics

1) Process Metrics
2) Product Metrics
3) Project Metrics.
   Quality
   Testing

Area

The most ~~ll~~ established area of Software metrics is cost and size estimation techniques.

The prediction of quality levels for software metrics is on reliability.

Problem during implementation :->

Statement :- Software development is to complex, it can't be managed like other parts of the organisation.

Management view :- forget it, we will find developers and managers who will manage that development.

1) Product Metrics :-

describes the characteristics of the product such as size, complexity, design features, performance, efficiency, reliability, portability etc.

2) Process Metrics :- describes the effectiveness and quality of the processes that produce the software product.

Example :-> • effort required in the product
• time to produce the product
• effectiveness of defect removal during development.
• No of defects found during testing
• maturity of the process

3) Project Metrics :-

describes the project characteristics and execution

Examples :- • Number of software developers
• Staffing pattern over the life cycle of software.
• Cost and Schedule.
• Productivity.

class

Date
Page

* Size Metrics

LOC :- "A line of code is any line of program text that is not a comment or blank line, regardless of the number of of statements or fragments of statements on the line. This specially includes all lines containing program header, declaration, execution and non-executable statements.

Advantages of using LOC as a size metric :-

- It is widely used and universally accepted.
- It permits the comparison of size and productivity metrics b/w diverse development groups.
- It directly relates to the end product
- LOC are easily measured upon project completion.
- It measures soft from the developers point of view. what he actually does (Write Loc)
- Continous improvement activities are base for estimation techniques.

Disadvantages :-

- LOC is language dependent. A line of C is not the same as a line of COBOL.

- It is not Consistent because same lines are more difficult to code than others.

- Sourca instructions vary with coding languages, design methods and with programmers ability.

- No ~~id~~ industry standard for measuring LOC

- LOC can't be used for normalizing if platforms and platforms are different.

* FP Metrics :-

- ~~functional~~ Count :- Alan Albert with IBM.

The function point is a unit of measurement to expre
the amount of business functionality an inform
System (of a product) provides to user.
Functional points are used to compute a functiona
size measurement of Software.

Advantage of using function Points as a size metric :

Because function point analysis is independent of language used, development platform etc. it can be used to identify the productivity benifits of ----- :
- one programming language over another
- one development platform methodology over another

- function points can be estimated from requirement specification or design specification, thus making it possible to estimate development efforts in early phases of development.
- function points are directly linked to statement of requirements

Disadvantages of using function points :

- function points counts are affected by project size
- Difficult to apply to massively distributed systems or to complex software system.
- Difficult to define logical files from physical files.
- validaty of weights - and consistancy of application has been changed.

Software project planning

In order to conduct a successfull software project. We must understand.

- Scope of work to be done
- The risk to be incurred
- The resource required
- The task to be accomplished
- The Cost to be extended
- The ~~sce~~ schedule to be followed.

software planning begins ~~fet~~ before technical work begin, continues as the software evolves from concept to reality and culminates only when the software is retired.

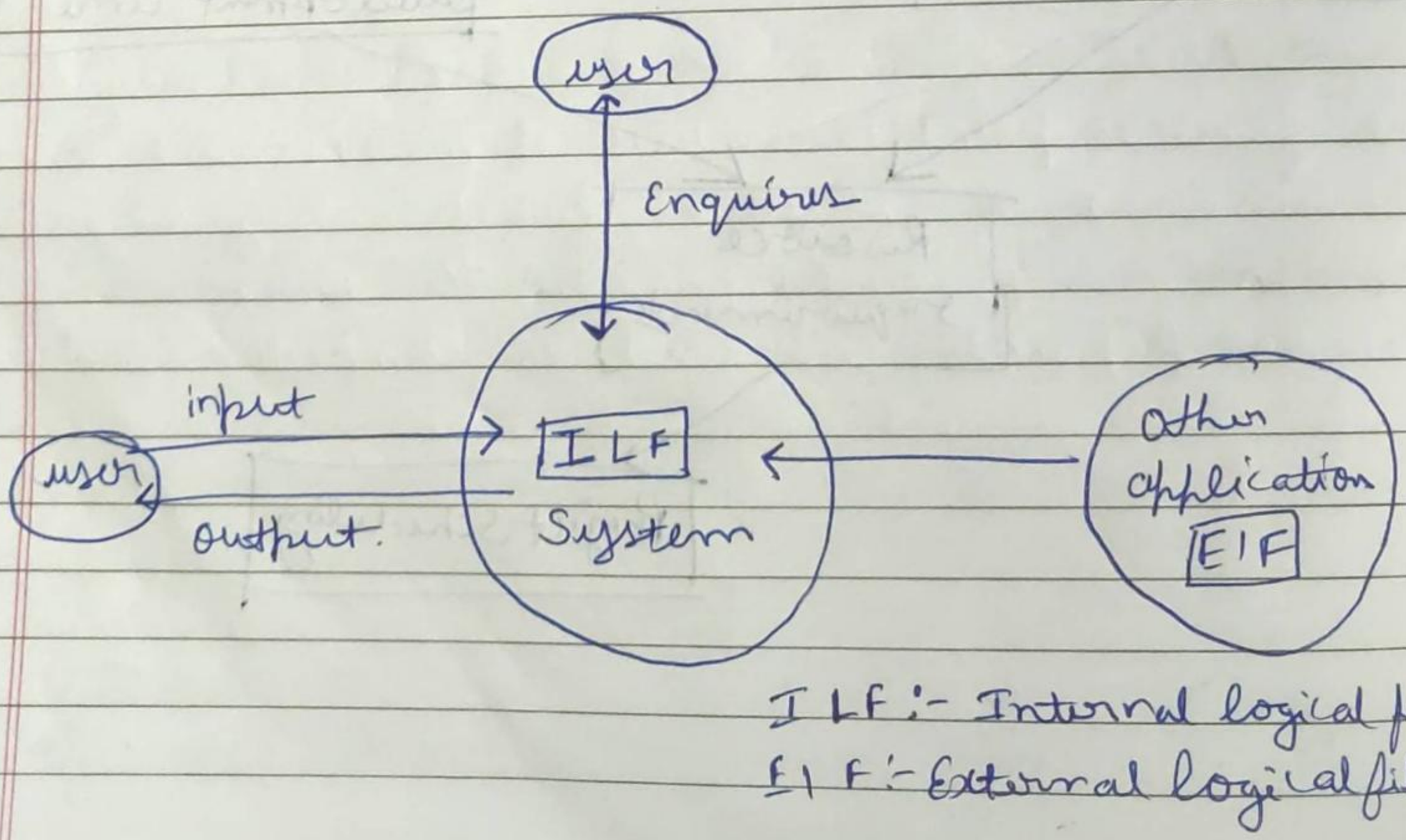
Activities during Software project planning.

Function Count (FPA - function point analysis) by Albrechit

→ Solution to size measurement problems

The principle of ~~Alba~~ Albrecht's function point analysis (FPA) is that a system is decompond into functional unit's.

- Inputs : information entering the system
- Outputs : information leaving the systems.
- Enquiras : requests for instant access to information
- Internal logical files : information held within system
- External logical files : information held by other systems that is used by the system being analyzed



ILF :- Internal logical files
EIF :- External logical files

Two functional types

1) A user identifable group of data
2) Transactional functional types

Counting Function Points

Functional units	Weighting factors		
	Low	Average (medium)	high
External Inputs [EI]	3	4	6
External Outputs [EO]	4	5	7
External Inquires [EQ]	3	7	6
Internal logical factor [ILF]	7	10	15
External Logical factor [EIF]	5	7	10

TCF → Technical Complexity Factor

functional units with weighting factor.

Organisations that use function point methods develop a criteria for determining whether a particular entry is low, Average or high.

$$FP = UFP * CAF$$

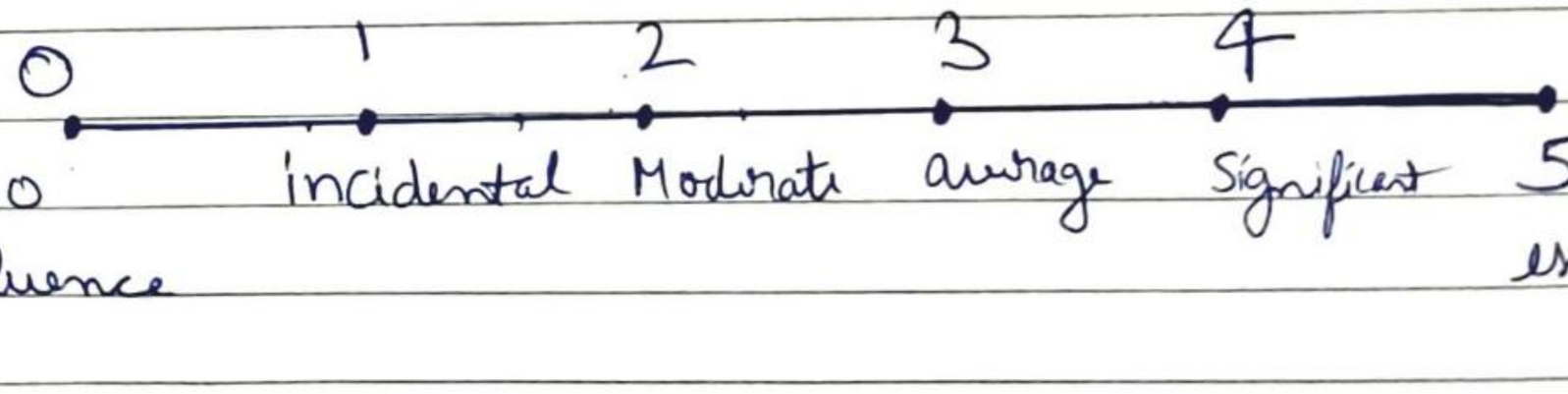
UFP :- Unadjustable function Point

TCF/ CAF :- Complexity Adjustment factor and is equal to $[0.65 + 0.01 \times \Sigma Fi]$

the F1/F2 Fi (i=1 to 14) are the degree of influence and are based on responses to questions noted in table.

UFP we require use of above table (Unadjusted function Point).

Rate each factor 0 to 5 on scale.



Example:
Consider a project with the following functional units
Number of user inputs = 50
Number of user outputs = 40
Number of user enquiries = 35
Number of user files (ILF) = 06.
Number of external interface = 04

1 Assume all complexity adjustment factors and weighting factors are average, Compute the function point for the project?

Solution:-

UFP ⇒ $50 \times 4 + 40 \times 5 + 35 \times 4 + 06 \times 10 + 4 \times 7$

UFP ⇒ 628

$$CAF \Rightarrow 0.65 + \left[0.01 \times \sum_{i=1}^{14} Fi\right]$$

⇒ $0.65 + [0.01 \times (14 \times 3)]$ → because average is taken

~~(3+3+3+3+.....+3) 14 times~~

⇒ 1.07

FP ⇒ 628×1.07 ⇒ 671.96

FP ⇒ 672 always Integer value.

8 Suppose a software project
I - 20 assuming first, factors / compon
O - 15 to be complex, 4th one is avera
In - 5 5th One is low.
usrfils - 5 Compute FP if out of 14 facto
EIF - 3 internal processing is very requir
, data Communicate is essential (5),
performance criticality is seriously issue,
reusability of Code is significant?
Online data entry is significant and the
rest factor are moderate?

Solution UFP ⇒ 20×6 + 15×7 + 5×6 + 5×10 + 3×
⇒ 120 + 105 + 30 + 50 + 15
⇒ 320

CAF ⇒ 0.65 + [0.01 × (9 × 2

Example :-

i) External Inputs.
a) 10 with low complexity
b) 15 with Avg
c) 17 with high

ii) External Output.
a) 6 with low
b) 13 with high

iii) External Inquiries :-
a) 3 with low
b) 4 with avg
c) 2 with high

iv) ILF
a) 2 with avg
b) 1 with high

v) EIF
a) 9 with low

(i) Significant (4)
(ii) Performance is very critical (5)
(iii) " " may be moderate (2)
(iv) System is not designed for multiple installation in diff (0). Other are average (3).

Solution :- $(10\times3 + 15\times4 + 17\times6) + (6\times4 + 13\times7) + (3\times3 + 9\times4 + 2\times6) + (2\times10 + 1\times15) + 9\times5$

$\Rightarrow 30 + 60 + 102 + 24 + 91 + 9 + 16 + 12 + 20 + 15 + 45$

$\Rightarrow 424$

$CAP \Rightarrow 0.65 + (0.01 \times (30 + 11))$

$\Rightarrow 0.65 + 0.41$

$\Rightarrow 1.06$

$\Rightarrow 424 \times 1.06$

$\Rightarrow 449.44$

$\Rightarrow 449$

Average Manning = $\frac{\text{effort (Cost)}}{\text{Duration}}$	No of person required for month the project
Project staffing	

$C = aL^b$

C — cost / effort
L — line of code
b — Constant

Cost

SFL Model ⇒ Cost = $1.4 \times L^{0.93}$ DOC ⇒ $30.4 + L^{0.90}$

Watsonfetix ⇒ Cost = $5.2 \times L^{0.91}$

Duration :-

SFL ⇒ $4.6 \times L^{0.26}$

Watson fetix ⇒ $4.1 \times L^{0.36}$

$$L = \left(\frac{C}{a}\right)^{1/b}$$

~~Productg~~

Productivity = $\frac{\text{LOC}}{\text{effort / Cost}}$

{LOC per Unit Cost = Productivity}

Q. Compare the SEL model and Watson Hafix model which is having 8 PY effort
a) Compute LOC for each model
b) Compute the duration.
c) productivity expressed as LOC/PY
d) Compute the avg staffing require for each model.

Ans:-

$$L = \left(\frac{C}{a}\right)^{1/b} = \left(\frac{C}{1.4}\right)^{1/0.93} \text{ KLOC} \Bigg\} \quad 92.264 \text{ KLOC}$$

$$L = \left(\frac{C}{5.2}\right)^{1/0.91} \text{ KLOC} \Bigg\} \text{ Watson} \quad \underline{24.632 \text{ KLOC}}$$

$= \text{POWER}(C/5.2, 1/0.91)$

In Excel $= \text{POWER}(A_1/5.2, B_1)$

b) Duration

SEL $\rightarrow 4.6 \times (94264)^{0.26} \Rightarrow 14.999 \approx 15$ months

WF $\rightarrow 4.1 \times (24.632)^{0.36} \Rightarrow 12.993 \approx 13$ months

c) productivity.

SEL $\Rightarrow \frac{\text{effort}}{\text{Duration}} \Rightarrow \frac{96}{15} \Rightarrow 6.4$

WF $\Rightarrow \frac{\text{effort}}{\text{Duration}} \Rightarrow \frac{96}{13} \Rightarrow 7.38$

COCOMO :- (Constructive Cost Model) :-

By Barry Boehm.

i) Basic COCOMO
ii) Intermediate COCOMO
iii) Detailed COCOMO

i) Basic

This model aims in ~~usta~~ estimating very quickly the cost as well as schedule for develop activity.
Now Soft projects (modes) are classified as organic, Semi detached and embeded.

Depending upon the model we have the following formulas :-

$$E = a.(KLOC)^b \ PM$$

E - effort
PM → Person Months

$$D = c(E)^d \ M$$

(a, b, c, d) are Constants which depends wether the Software project is organic, semi detached, embedded.
D → Duration
M → Months

~~Organic~~

Organic Effort = $2.4(KLOC)^{1.05}$ PM
Semi-detach Effort = $3.0 \ (KLOC)^{1.12}$ PM
Embeded Effort = $3.6 \ (KLOC)^{1.20}$ PM

Organic Tdev = $2.5 \ (Effort)^{0.38}$ months
Semi-detach Tdev = $2.5 \ (effort)^{0.35}$ months
Embeded Tdev = $2.5 \ (effort)^{0.32}$ months

Values are a, b, c, d for basic Cocomo model are :-

Project	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embeded	3.6	1.20	2.5	0.32

Q Compute effort, TDEV, average staffs, productivity for SP of size 200 KLOC.

i) Effort :- $a \cdot (KLOC)^b$

$= 2.4\ (200)^{1.05} \Rightarrow 1133.12\ PM$

ii) Duration $\Rightarrow 2.5\ (1133.12)^{0.35} \Rightarrow 29.3$

iii) Avg Staff $\Rightarrow \frac{1133.12}{29.3} \Rightarrow 31.67$ person

iv) Productivity $\Rightarrow \frac{200}{1137.12} \Rightarrow 0.1769\ KLOC/PM$

Organic :- It belongs to ;

* Intermediate Modul :-

There are four categories of Cost drivers

i) Product attributes :-
   - RELY
   - DATA
   - CPLX

II) Computer Attributes :-
   - Time
   - STOR
   - VIRT
   - TURN

III) Personal Attributes :-
   - MODP
   - TOOL
   - SCED

RELY	Required reliability
DATA	Database Size
CPLX	Product complexity
TIME	Execution time Constraint
STOR	Main Storage Constraint
VIRT	Virtual Machine Constraint.
TURN	Computer turnaround time
ACAP	Analyst Capability
AEXP	Application Experience
PCAP	Programmer Capability
VEXP	Virtual Machine Experience
LEXP	Programming language Experience.
MODP	Modern programmer practices.
TOOL	Use of Soft tool.
SCED	Required development schedule.

* ~~Visula~~ Visualisation tool / Project scheduling tool.

Grantt :-

PERT :- programming evaluation and review technique

Cost Drivers	RATINGS					
	Very Low	Low	Normal	High	very High	Extra High
I) Product						
1. RELY	0.75	0.88	1	1.15	1.40	—
2. DATA	—	0.94	1	1.08	1.16	—
3. CPLX	0.70	0.85	1	1.15	1.30	1.65
II) Computer						
4. TIME	—	—	1	1.11	1.30	1.66
5. STOR	—	—	1	1.06	1.21	1.56
6. VIRT	—	0.87	1	1.15	1.30	—
7. TURN	—	0.87	1	1.07	1.15	—
III) Personal						
8. ACAP	1.46	1.19	1	0.86	0.71	—
9. AEXP	1.29	1.13	1	0.91	0.82	—
10. PCAP	1.42	1.17	1	0.86	0.70	—
11. VEXP	1.21	1.10	1	0.90	—	—
12. LEXP	1.14	1.07	1	0.95	—	—
IV) Project						
13 MODP	1.24	1.10	1	0.91	0.82	—
14 TOOL	1.24	1.10	1	0.91	0.83	—
15 SCED	1.23	1.08	1	1.04	1.10	—

* A new project with 400KLOC embedded system has to be develop. Project management has a choice of ~~having~~ hiring ~~project management has a~~ from three pools of developers.

a) very high capable with very little experience in programming language being used

b) developers of low quality with a lot of experience with the programming language,

c) Developers who have high programming language experience along with very high uses of software tools

If you are a project manager who has to hire, which team will you hire and why?

OR

What is the impact of having all developers from one or the other pool and decide which is your optimal choice?
Also compute the staffing size and productivity of the project for the team finally choice chosen by you?

Project	a	b	c	d
Organic	3.2	1.05	2.5	0.38
Semi detached	3.0	1.12	2.5	0.35
Embedded	2.8	1.20	2.5	0.32

Contd

- for intermediate COCOMO model we first compute the initial effort which is computed as follows.

$$E_i = a \times (KLOC)^b$$

- Next we compute the value of EAF (effort adjustment factor) which simply the product of cost drivers.

3 finally we compute cost/effort as follows

$$E = EAF * E_i$$

Solution

$$E_i = 2.8 (400)^{1.2}$$
$$= 3712 \text{ PM}$$

Team A :- EAF => $0.70 \times 1.18 = 0.798$

$E \Rightarrow EAF * E_i \Rightarrow 3712 \times 0.798$

$\Rightarrow 2962.17 \text{ PM}$

$D \Rightarrow 2.5 (2962.17)^{0.32}$

$\Rightarrow 32.27 \text{ M}$

Team B :- EAF => $1.13 \times 0.95 \Rightarrow 1.0735$

$E \Rightarrow 3984.83 \text{ PM}$

$D \Rightarrow 2.5 (3984.83)^{0.32} \Rightarrow 35.987 \text{ M}$

Team C :- EAF :- $0.86 \times 1.13 \times 0.95 \times 0.87$

$= 0.766$

$E \Rightarrow 2894.37 \text{ PM}$

$D \Rightarrow 2.5 (2894.37)^{0.32}$

$D \Rightarrow 31.86 \text{ M}$

He can see that Team C requires minimum amount of cost and time.

Here,

as the project manager I could decide to hire third team of developers who have high analyst capability low experience, high programming and very high use of software tools.