

Process Metrics - are collected across all projects & over long periods of time. Its intent is to provide a set of process indicators that lead to long term software improvement.

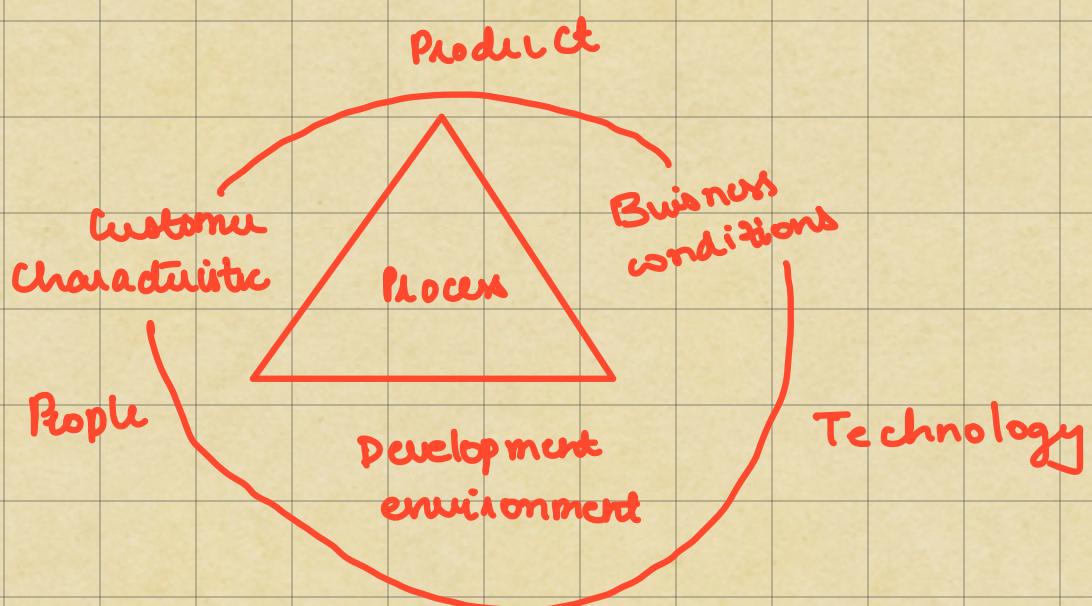
→ Rational way to do : measure specific attr. of a process



develop meaningful metric on these
attributes



Use these to develop indicators



Software Measurement

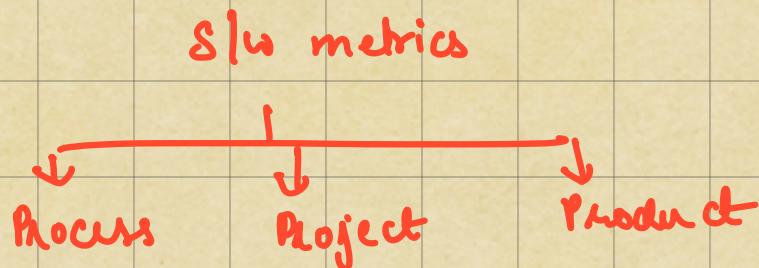
- Direct measures

Software : cost + effort applied

Product : LOC, memory size, execution speed, defects reported over a set period of time.

- Indirect measures

Product : quality, complexity, efficiency, functionality, reusability, maintainability & other abilities



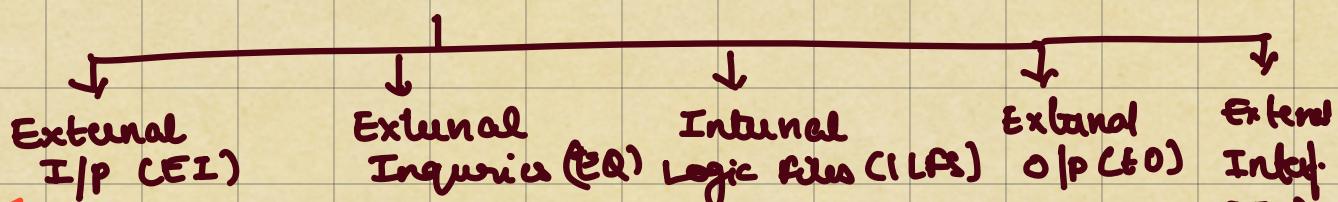
Functionality → Function Point Metrics

↳ user's POV

(user requests & receives in return)

- i) estimate cost or effort for coding, design, or testing (FP tells size of functionality → ↑ → more coding, design, testing)
- ii) Predict the no. of error that will be encountered in testing (historical data)
- iii) forecast no. of components, no. of projected source lines in the implemented system (FP → LOC depending on programming lang.)

Function-Based Metrics



• provided by user/
transmitted by system
• data/control info coming
from outside the appn.

Logical grouping of data residing within appn.
• data sent outside useful data
• derived data to app

boundary
 ↙ ↙
 some online i/p
 that result in
 the immediate generation
 of some online o/p
 (not IFS)

within appn.
 boundary

that provides
 info to user
 (report, success,
 error)

components xx

IDV	Count	Weighting factor		
		S	A	C
EI	x	3	4	6
EQ	x	4	5	7
EO	x	3	4	6
ILF	x	7	10	15
EIFS	x	5	7	10

$$\frac{\text{count}}{\text{total}} =$$

$$FP = \text{Count total} \times [0.65 + 0.01 \times (\Sigma f_i)]$$


 Value Adjustment Factor

Q.

I D N	S	A	C	
E I	12	x	3	4 6
E O	24	x	4	5 7
E Q	30	x	3	4 6
ILFs	4	x	7	10 15
EIFs	6	x	5	7 10
				<u>30</u>
				280 (count total)

user form details etc.

External Input → what data comes into the system?

notif, reports, ticket, invoices, dashb

External Output → what processed data comes out

of the system?

search funcn, status check

External Inquiries → Information retrieved w/o

updating data

core db, history

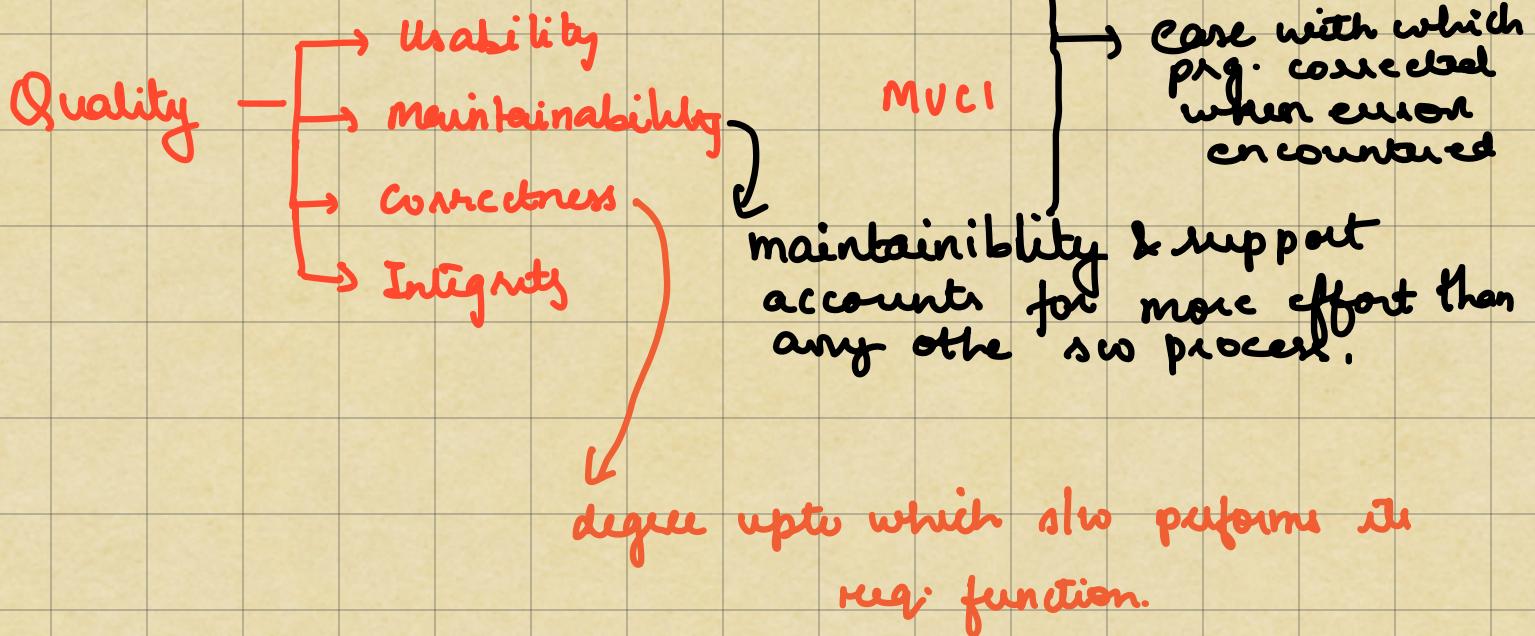
ILF → What does the system store & maintain internally

payment gateway, third party APIs

ELF → What external data the system use but not store?

party / vendor db's

→ enhanced it
customer and
adapt to



defects / KLOC

↳ lack of conformance to req.
reported by user after released for gen. use
counted over std. period of time

MTTC (mean time to change) → analyze
request, design app. modif., implement, test,
distribute it to all users

Integrity : measures the system's ability to withstand attacks (accidental / intentional) to its security.

Attacks → prog., data, documentation

Two addn. attributes

- ↳ Threat: prob. that an attack of a spec. type will occur within a given time.
- ↳ Security: prob. that an attack of a 1/s type will be repelled.

$$\text{Integrity} = \sum [1 - (\text{threat} \times (1 - \text{security}))]$$

$$= 0.0125$$

$$\alpha \cdot \text{threat} = 0.25$$

$$\text{security} = 0.95$$

$$= \sum [1 - (0.25 \times (1 - 0.95))] \\ = 0.99$$

$$\alpha \cdot \text{threat} = 0.5$$

$$\text{security} = 0.25$$

$$I = [1 - (0.5 \times (1 - 0.25))] \\ = 0.63$$

Usability: if a program is not easy to use it is often doomed to failure, even if the functions it performs are valuable.

helps us measure the falling ability of quality
DRE (Defect Removal Eff.) assurance + control activities applied
 ↳ project + process. throughout framework.

$$DRE = \frac{E}{E+D}$$

↑ no. of errors found bfr.
delivery of sw

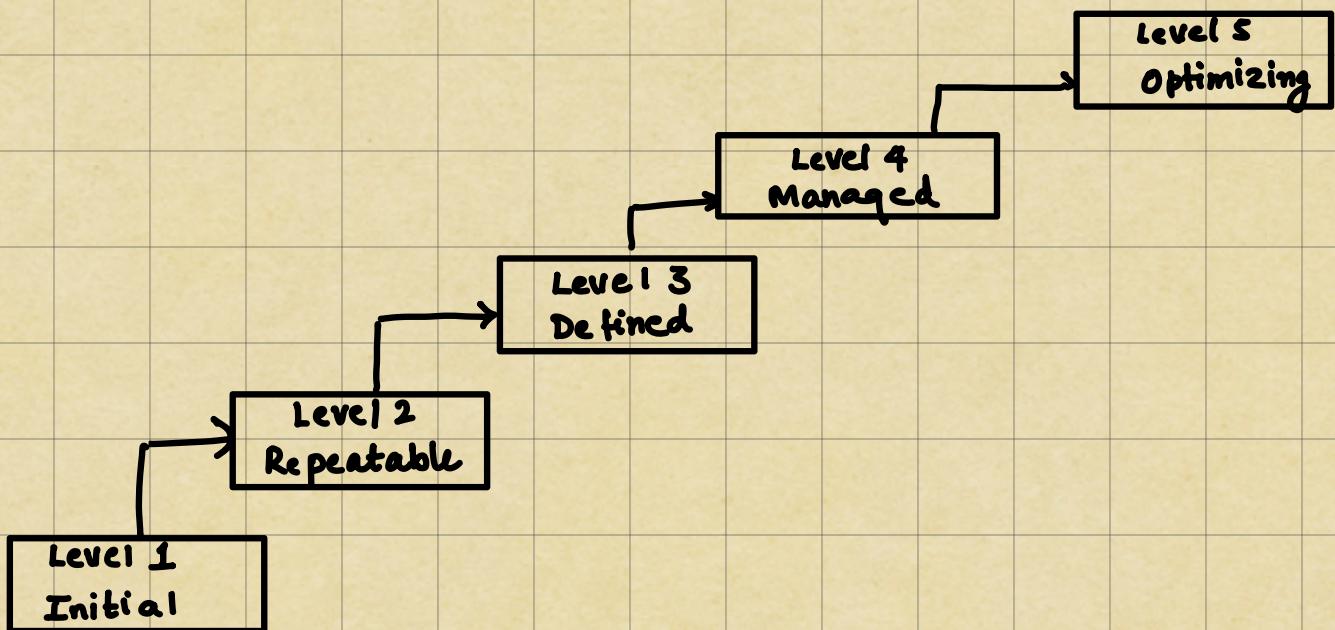
↑ no. of defect found after
delivery

ideal value = 1

$$E \uparrow \Rightarrow D \downarrow$$

CMM (Capability Maturity Model)

- focus is on the process of developing a software
- there are 5 levels → a company estimates which level their dev. process (higher the better)
- CMM model also helps in jumping to the next level.



Level 1: → software dev. process is unpredictable wrt to time & cost.

→ dependent on team as team changes process changes

Level 2: → focuses on establishing basic project management policies

→ Experience with prev. proj is used for managing new similar natured proj.

Level 3: → Process of developing the proj is documented

→ Training programs : to enhance the skills & knowledge of team members to ↑ efficiency

Level 4: → focuses on quantitative management of processes

→ Sets quantitative goals for slw quality.

Level 5: → focuses on cont. iterative improvement

→ Organizations use feedback & innovative tech to improve overall processes.