PAGE NO.

Pages: 4

Date: 21/10/20
Roll Number: 17010208I

Software Measurement

Direct measures → length
Indirect measures → quality of software

Direct Measure

(LOC) → Line of Code
Speed of execution, amount of money used, number of defects reported.

Indirect measures

→ "abilities" of the software
→ Functionality
→ Complexity
→ Reliability
→ Robustness

Size Oriented Metrics

It is computed by normalizing quality and/or productivity measures while considering the size of the software generated

Project	lines of code	effort (person months)	money (1000 $)	pages of document	errors	defects	people involved
α	1200	24	168	365	134	29	3
β	2700	62	440	1200	320	86	5
γ	20000	40	314	1000	256	60	6

We can derive errors/LOC, Defects/LOC, $(money)/LOC and pages of document/LOC.

→ Usage of LOC as a normalizing constant is debatable

→ Software measurement can be categorized in two ways :-

a) Direct Measures → This includes measurement of cost and effort applied. Direct measures of product include lines of code (LOC) produced, execution speed, memory size and defects reported over some specified period of time.

b) Indirect Measures → This include functionality, quality, complexity, efficiency, reliability, maintainability.

Size Oriented Metrics

They are derived by normalizing quality and/or productivity measures by considering the size of the software that has been produced.

From the rudimentary data contained in the previous table, a set of simple size-oriented metrics can be developed for each project:

- → Errors per KLOC (thousand lines of code)
- → Defects per KLOC
- → $ per LOC
- → Page of documentation per KLOC.

a) Size-Oriented metrics are not universally accepted as the best way to measure the process of S/W development.

b) Most of the controversy is about using 'LOC' for normalizing.

c) Proponents claim that 'LOC' is an artifact of all S/W projects that can be easily counted

d) Opponents argue that 'LOC' measures are programming language dependent and they penalize well-designed but short programs

DATE
PAGE NO.

that they can't easily accomodate non-procedural languages and that their use in estimation requires a level of detail that may be difficult to achieve.