

**Unit: III**  
**Lecture: 6**

**Function Oriented Metrics:**

Measuring software size in terms of lines of code is analogous to measuring a car stereo by the number of resistors, capacitors and integrated circuits involved in its production. The number of components is useful in predicting the number of assembly line staff needed but it does not say anything about the functions available in the finished stereo. When dealing with customers, manufacturer talks in terms of functions available and not in terms of components.

**Function Oriented Metrics measures the functionality from user point of view i.e. on the basis of what the user requests and receives in return.**

**Function Point (FP) is a function oriented metric suggested by Albrecht.**

In function points, the system functionality is calculated in terms of the number of functions it implements, the number of inputs, the number of outputs etc. parameters that can be obtained after requirement analysis and that are independent of the specification and implementation language.

Measurement parameter	Count	Weighting factor			=	[ ]
		Simple	Average	Complex		
Number of user inputs	[ ] ×	3	4	6	=	[ ]
Number of user outputs	[ ] ×	4	5	7	=	[ ]
Number of user inquiries	[ ] ×	3	4	6	=	[ ]
Number of files	[ ] ×	7	10	15	=	[ ]
Number of external interfaces	[ ] ×	5	7	10	=	[ ]
Count total	→					[ ]

Fig computing Function Point

Function points are computed by completing the table shown in fig above.

Five information domain characteristics are determined and counts are provided in the appropriate table location.

Information domain values are defined in the following manner:

- **Number of user inputs:** Each user input that provides distinct application oriented data to the software is counted.
- **Number of user outputs:** Each user output that provides application oriented information to the user is counted. In this context, output refers to reports, screens, error messages etc.
- **Number of user inquiries:** An inquiry is defined as an on-line input that results in the generation of some immediate software response in the form of an on-line output. Each distinct inquiry is counted.
- **Number of files:** Each logical master file (i.e. a logical grouping of data that may be one part of a large database or a separate file) is counted.
- **Number of external interfaces:** All machine readable interfaces (e.g. data files on storage media) that are used to transmit information to another system are counted.

Once these data have been collected, a complexity value is associated with each count. Organizations that use function point methods develop criteria for determining whether a particular entry is simple, average or complex.

To computer function points (FP), the following relationship is used:

$$\mathbf{FP = count\ total * [0.65 + 0.01 * \Sigma(F_i)]}$$

where count total is the sum of all FP entries obtained from fig above.

the  $F_i$  ( $i = 1$  to 14) are complexity adjustment values based on responses to 14 questions like:

1. Does the system require reliable backup and recovery?
2. Are data communication required?
3. Is performance critical?
4. Does system require on-line data entry? etc.

Each of such questions is answered using a scale that ranges from 0 (not important or applicable) to 5 (absolute essential).

**Advantages of FP:**

- FP is programming language independent, making it ideal for applications using conventional and nonprocedural languages.
- It is based on data that are more likely to be known early in the evolution of a project, making FP more attractive as an estimation approach.

**Disadvantages of FP:**

- The method requires some “sleight of hand” in that computation is based on subjective rather than objective data.
- FP has no direct physical meaning- its just a number.