6. Function points

It is a method that will allow Software Engineers to evaluate the dimension, time and cost (money) of software process.

6.1 Two SW metrics categories:

Direct:

- direct you can measure among them, e.g.: LOC, McCabe Index (to measure the complexity, how much cycles, decisions point),number of transactions (FP), OO (research...)

Indirect:

- You can derive them by basing on the service level agreement or those that are considering the users' opinion (quality of Software)

6.2 Dimension metrics

Based on the dimension of code, such as LOC.

LOC

Lines of code

seems easy to compute but is strictly associated with the actual programming language and programming style.

KLOC = thousands of line of code

6.3 Function point

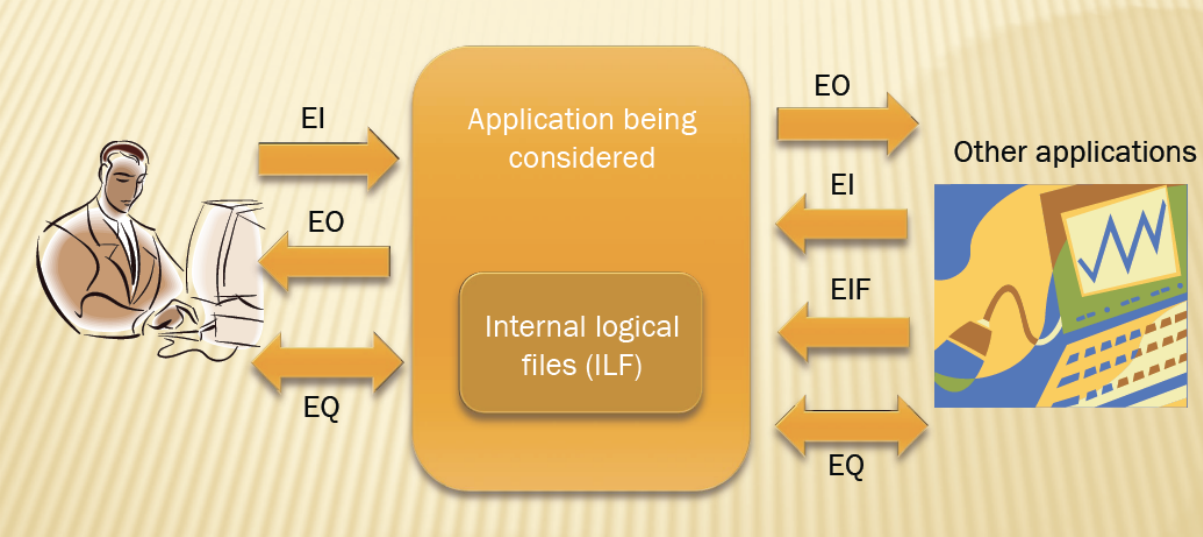
old ('79) empirical formula based on functionalities, several extensions to the original proposal

Functionality		Count	Weight			
EI		X	3-4-6		=	
EO		X	4-5-7		=	
EQ		X	3-4-6		=	
ILF		X	7-10-15		=	
EIF		X	5-7-10		=	
				Total		

$$FP = \text{Total} \times \left(0.65 + 0.01 \times \sum_{i=1}^{14} F_i\right)$$

Dimension of the sw.

6.4 Five functionalities of FP



ILF: An ILF is a user-identifiable group of logically related data or control information maintained within the boundary of the application. The primary intent of an ILF is to hold data maintained through one or more elementary processes of the application being counted.

EIF: An external interface file (EIF) is a user identifiable group of logically related data or control information referenced by the application, but maintained within the boundary of another application. The primary intent of an EIF is to hold data referenced through one or more elementary processes within the boundary of the application counted. This means an EIF counted for an application must be in an ILF in another application.

EI: An external input (EI) is an elementary process that processes data or control information that comes from outside the application boundary. The primary intent of an EI is to maintain one or more ILFs and/or to alter the behavior of the system.

EO: An external output (EO) is an elementary process that sends data or control information outside the application boundary. The primary intent of an

external output is to present information to a user through processing logic other than, or in addition to, the retrieval of data or control information . The processing logic must contain at least one mathematical formula or calculation, create derived data maintain one or more ILFs or alter the behavior of the system.

EQ: An external inquiry (EQ) is an elementary process that sends data or control information outside the application boundary. The primary intent of an external inquiry is to present information to a user through the retrieval of data or control information from an ILF of EIF. The processing logic contains no mathematical formulas or calculations, and does not create derived data.
No ILF is maintained during the processing, nor is the behavior of the system altered

Complexity/weights

Each functionality has a weight: Low, Medium, High

6.5 Adjusted FP

can change +/-35% using indicators and weights:

$$\text{AFP} = \text{Total} \times \left(0.65 + 0.01 \times \sum_{i=1}^{14} F_i\right)$$

AFP has been substituted, it's not used anymore.

6.6 FP limits

PROS:

widely used, certified personnel available, obj calculation, UFP independent of tech, can be used early in dev process, accurate as SLOC

CONS:

semantic difficulty, incompleteness, lack of automatic count, different version

6.7 Component of data functionalities

DET: user identifiable single field within a ILF/EIF

RET: User identifiable group of fields

The complexity of ILF or EIF is associated with the number of RET/DET

Ret/Det	1-19 Det	20-50 Det	51+ Det
1 Ret	Low (7/5)	Low (7/5)	Medium (10/7)
2-5 Ret	Low (7/5)	Medium (10/7)	High (15/10)
6+ Ret	Medium (10/7)	High (15/10)	High (15/10)

6.8 Transaction identification

Actions	EI	EO	EQ
1) Validate	can	can	can
2) Math calculations	can	must*	cannot
3) Select data using specific criteria	can	can	can
4) Evaluate Boolean conditions	can	can	can
5) Update one or more ILF	must* / p.goal	must*	cannot
6) Read one or more ILF /EIF	can	can	must
7) Get some control data	can	can	must
8) Compute new data	can	must*	cannot
9) Change the system behavior	must*/p.goal	must*	cannot
10) Find data and present them outside the application boundary	can	must/p.goal	must/p.goal
11) Acquire (control) data produced outside the application boundary	must	can	can
12) Manipulate data (e.g., sort)	can	can	can

Transaction components

Evaluate the complexity over the transactions:

FTR: count how many FTR have been changed

DET: single field with a ILF

Transaction complexity

EI

FTR / DET	1-4 DET	5-15 DET	16+ DET
0 – 1 FTR	Low (3)	Low (3)	Medium (4)
2 FTR	Low (3)	Medium (4)	High (6)
3+ FTR	Medium (4)	High (6)	High (6)

EO/EQ

FTR / DET	1-5 DET	6/19 DET	20+ DET
0 – 1 FTR	Low (4/3)	Low (4/3)	Medium (5/4)
2-3 FTR	Low (4/3)	Medium (5/4)	High (7/6)
4+ FTR	Medium (5/4)	High (7/6)	High (7/6)

SEE EXAMPLES IN SE_10_FUNCTIONPOINTS AND SE_10_VIDEOONLINE