Politechnika Wrocławska

Software Project in Management

2019/2020

dr Sebastian Tomczak

Estimating the complexity of the project

FUNCTION POINT (FP)

SOFTWARE SIZE MEASURES

- **Number of physical source code lines (SLOC) -** lines containing instructions and declarations are taken into account. A line containing several instructions or declarations is only considered once. Lines empty or containing only one comment are skipped.

- **Function points -** are associated with specific types of software functionality.

- **Construction points -** assigned to construction elements.

FUNCTIONALITY MEASURES

- The metrics of functional points: over 20 versions
  - FP (Function Point) – Albrecht
  - MARK II (Symons)
  - IFPUG
  - NESMA
  - SPR (Software Productivity Research)
  - Points of objects
  - The number of UCP (Use Case Points)
  - The metric Bang DeMarco – appropriateness of set of functions in relation to objectives and user's tasks user's
  - Points of the web – object ones
  - Story points
  - COSMIC
  - …
- The number of functions performed by the programme
- The number of items of the functional structure
- The number of items of the informative structure
- The accuracy of the obtained result

FUNCTION POINT METHOD

- Function points (FP) are understood as a measure of the size of computer applications and design that must be created.
- It is a measure created mainly for the purpose of estimating the size and costs of the project, which, for example, we negotiate with the client at the initial stage of the project.

- The basis for measuring is functionality planning.

- It is independent of the programming language, development methodology, technology or the ability of the project groups used to create the application.

FUNCTION POINT METHOD

- The FP method is not an excellent measure of the workload of creating an application or valuation of its business value, although the size of the project given in function points is an important factor in measuring each of these two values.

- Example:
  - The cost of building a house A (150 m2) is usually less than the cost of building a house B (450 m2). However, the materials used: marble bathrooms, fittings and floors can make a smaller house more expensive. Other factors such as location and number of bedrooms can also make a smaller home a more valuable home.

FUNCTION POINT METHOD

1. A measure of productivity
2. Support development estimation
3. Monitoring of external contracts (Outsourcing)
4. Help in business decisions
5. Normalization of other measures

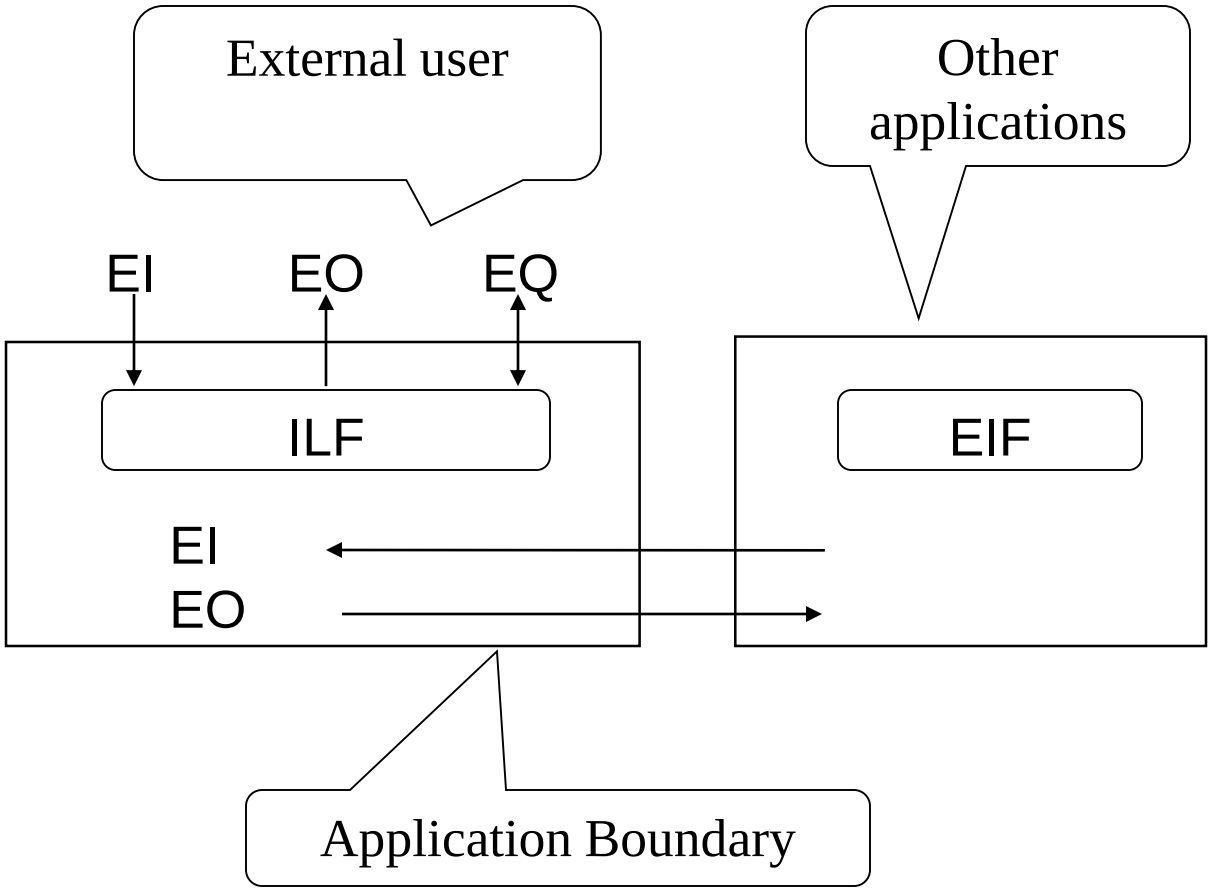
FUNCTION POINT METHOD

FUNCTION POINT ANALYSIS CLASSES

- Development project function point count
- Enhancement project function point count
- Application function point count

FUNCTION POINT METHOD

COUNTING COMPONENTS



FUNCTION POINT METHOD

COUNTING COMPONENTS

- Informative
  - ILF – Internal Logic File
  - EIF – External Logic File
- Functional
  - EI – External Input
  - EO - External Output
  - EQ – External Inquiry

FUNCTION POINT METHOD

TYPES OF INFORMATION COMPONENTS

- Data – sets of events or objects being stored in memory and processed, as for ILF within the application, and as far as EIF is concerned by other application (provider, receiver, ware, invoice)

- Control information – data used by the application to influence the elementary process / function. These are rules or parameters being stored in memory and maintenance, as for ILF within the application, and as far as EIL is concerned by the other application

- Data and control information are identified as user's requirements

FUNCTION POINT METHOD

COUNTING COMPONENTS

- ILF – is a user identifiable group of logically related data or control information maintained through an elementary process of the application within the boundary of the application.

- EIF – is a user identifiable group of logically related data or control information referenced by the application but maintained within the boundary of a different application.

- Each identified ILF and EIF must be assigned a functional complexity based on the number of data element types (DET) and (RET) associated with the ILF or EIF.

FUNCTION POINT METHOD

TYPES OF FUNCTIONAL COMPONENTS

- An elementary process / function of an application – the process of handling of data or control information (C,U,D,R,A)

- Logics of maintenance each process / function must be unique

FUNCTION POINT METHOD

COUNTING COMPONENTS

- EI – is the elementary process of the application which processes data or control information that enters from outside the boundary of the application.

- EO – is the elementary process of an application which generates data or control information that exits the boundary of the application.

FUNCTION POINT METHOD

COUNTING COMPONENTS

- EQ – is an elementary process of the application which is made up of input-output combination that results in data retrieval.
- The input side is the control information which spells put the request, specifying what and/or how data is to be retrieved.
- The output side contains no derived data. No ILF is maintained during processing.

FUNCTION POINT METHOD

STAGES

1. Calculation of the „not-fitted” functional points
2. Calculating the indicator of the level technical complexity
3. Calculating the system size by means of the number of functional points

FUNCTION POINT METHOD

STAGE 1

a) Identify the application boundary

b) Classification of system's constituents according to functional types

c) Determining the level of complexity for every constituent of the given functional type on the basis of the complexity matrix

FUNCTION POINT METHOD

STAGE 1

d) Calculation of UFP according to the formula:

$$UFP = \sum_{i=1}^{5}\sum_{j=1}^{3} w_{ij} z_{ij}$$

z_{ij} – the number of constituents of the ith type and the jth complexity

w_{ij} – the weight of the z_{ij} constituents

$i = 1, ...,5$

$j = 1, 2, 3$

FUNCTION POINT METHOD

WEIGHTS OF FUNCTIONAL TYPES

Type	Level of complexity		
	L	A	H
ILF	7	10	15
EIF	5	7	10
EI	3	4	6
EO	4	5	7
EQ	3	4	6

L – low A – average H - high

FUNCTION POINT METHOD

CONSTITUENTS OF TYPES SYSTEM'S ELEMENTS

- DET – Data element types
- RET – Record element types
- FTR – File types referenced

FUNCTION POINT METHOD

ILF AND EIF COMPLEXITY MATRIX

The RET number	The DET number		
	1 - 19	20 - 50	51+
<2	L	L	A
2 - 5	L	A	H
>5	A	H	H

FUNCTION POINT METHOD

RULES OF CALCULATING THE COMPLEXITY OF ILF AND EIF

- The DET number – the number of attributes / fields along with external keys
- The RET number taking account of the hierarchic relations
- No types of the objects / files being created due to technological reasons (e.g. separate text files) are being included in the RET

FUNCTION POINT METHOD

EI COMPLEXITY MATRIX

The FTR number	The DET number		
	1- 4	5 - 15	16+
<2	L	L	A
2	L	A	H
>2	A	H	H

FUNCTION POINT METHOD

RULES GOVERNING THE CALCULATION OF EI AND EO COMPLEXITY

- The DET number – the number of attributes / fields being processed aling with control information
- One adds error messages as well as hints on o the DET
- The FTR number – the number of ILF, EIF participating in processing
- The FTR number taking into account the following hierarchical relations ILF, EIF

FUNCTION POINT METHOD

EO COMPLEXITY MATRIX

The FTR number	The DET number		
	1- 5	6 - 19	20+
<2	L	L	A
2 - 3	L	A	H
>3	A	H	H

FUNCTION POINT METHOD

EQ COMPLEXITY MATRIX – THE SIDE OF INPUT

The FTR number	The DET number		
	1- 4	5 - 15	16+
<2	L	L	A
2	L	A	H
>2	A	H	H

FUNCTION POINT METHOD

EQ COMPLEXITY MATRIX - THE SIDE OF OUTPUT

The FTR number	The DET number		
	1- 5	6 - 19	20+
<2	L	L	A
2 - 3	L	A	H
>3	A	H	H

FUNCTION POINT METHOD

RULES GOVERNING CALCULATION OF EQ COMPLEXITY

- The quantity DET of input – attributes / fields along with control information, which cut across the application's border and these are situated on the input's side of the query
- The DET number for output - attributes / fields along with control information, which cut across the application's border and these are situated on the output's side of the query
- One add up error messages, hints to the DET number
- The FTR number – the number of ILF, EIF connected with the input and output
- EQ complexity - max (DET in,DET out)

FUNCTION POINT METHOD

STAGE 2

a) Establishing the level and the number of degrees of influence of each characteristics on the basis of the table of rules

b) Calculating the total degree of influence

$$DI = \sum c_i$$

$$i = 1, ...,14$$

c) Calculating the indicator of technical complexity

$$TCF = 0.65 + 0.01DI$$

FUNCTION POINT METHOD

THE LIST OF CHARACTERISTICS OF SYSTEM'S TECHNICAL COMPLEXITY

ID	Distinguishing feature	ID	Distinguishing feature
C1	Data Communications	C8	On-Line Update
C2	Distributed Data Processing	C9	Complex Processing
C3	Performance	C10	Reusability
C4	Heavily Used Configuration	C11	Installation Ease
C5	Transaction Rate	C12	Operational Ease
C6	On-line Data Entry	C13	Multiple Sites
C7	End-User Efficience	C14	Facilitate Change

FUNCTION POINT METHOD

DEGREE OF INFLUENCE (DI)

Influence	Number of c_i degrees
Not present, or no influence	0
Incidental influence	1
Moderate influence	2
Average influence	3
Significant influence	4
Strong influence throughout	5

FUNCTION POINT METHOD

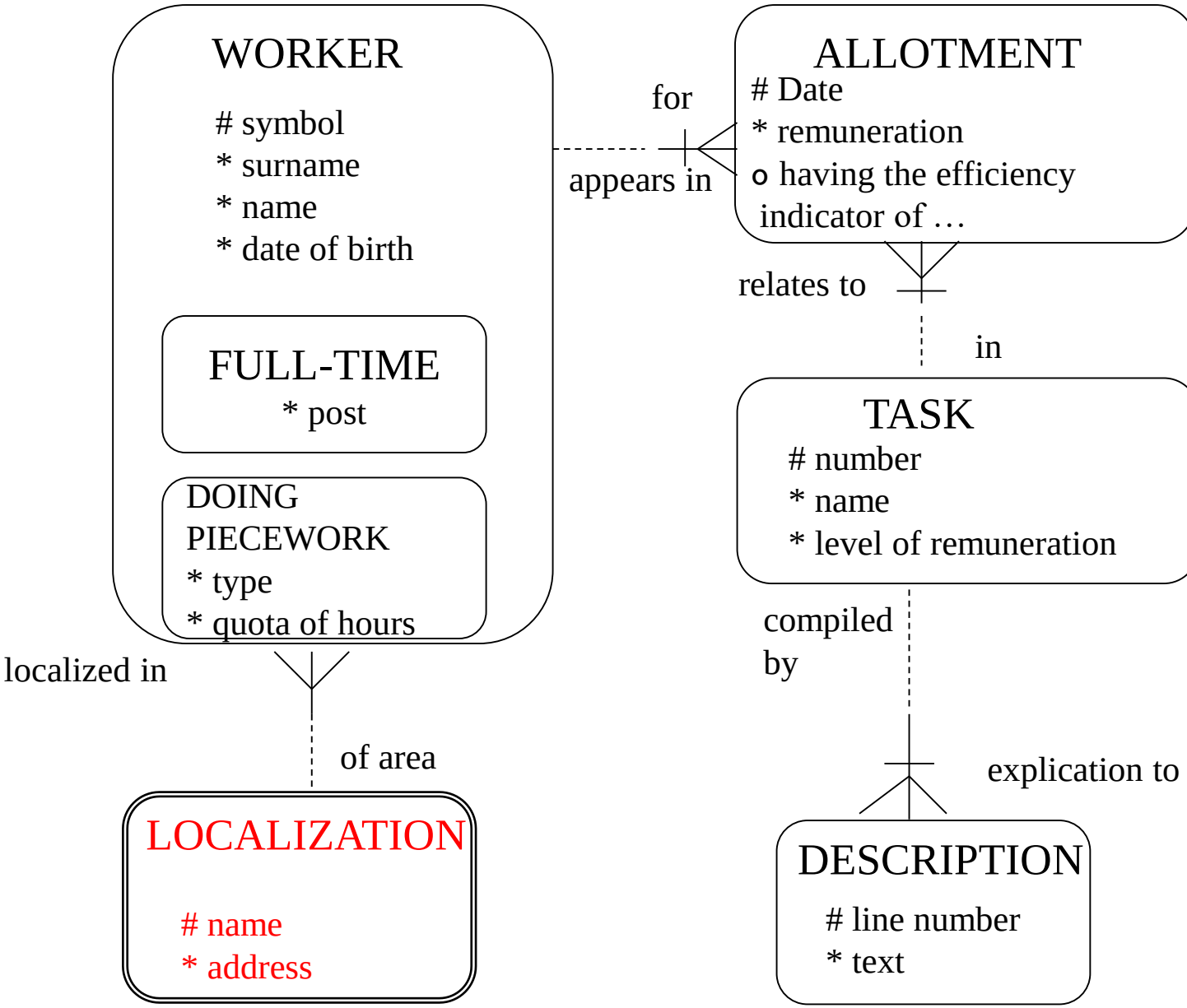
RULES OF DETERMINIG THE INPUT DEGREE OF THE C1 CHARACTERISTICS

DI number	Description of requierements
0	Application is pure batch processing or stand alone PC.
1	Application is pure batch but has remote data entry or remote printing.
2	Application is pure batch but has remote data and remote printing.
3	On-line data collection or TP (teleprocessing) front end to a batch process or query system .
4	More than a front-end, but the application supports only one type of TP communications protocol .
5	More than a front-end, but the application supports more than one type of TP communications protocol.

FUNCTION POINT METHOD
STAGE 3

$$FP = UFP * TCF$$

Example


WORKER
symbol
* surname
* name
* date of birth
FULL-TIME
* post
DOING PIECEWORK
* type
* quota of hours
for
appears in
ALLOTMENT
Date
* remuneration
o having the efficiency indicator of …
relates to
in
TASK
number
* name
* level of remuneration
compiled by
explication to
DESCRIPTION
line number
* text
localized in
of area
LOCALIZATION
name
* address

LIST OF ELEMENTARY FUNCTIONS IN THIS EXAMPLE

Z111. Hire a new worker
Z113. Promote the worker
Z115. Sack the worker

Z213. Assign the worter to a task

Z311. Evaluate the worker by means of the efficiency indicator

Z411. Draw up a report on worker's localizations
Z421. Determine tasks assigned to the worker … for the working day of …

TABLE OF CONNECTIONS BETWEEN FUNCTION AND OBJECT

	WORKER	TASK	ALLOTMENT	LOCALIZATION
Z111	C			R
Z113	R/U			
Z115	D/A	R	D/A	
Z213	R	R	C	
Z311	R	R	R/U	
Z411	R			R
Z421	R	R	R	

STAGE 1

Name of constituent	Type	The RET/FTR number	The DET number	Complexity
WORKER ALLOTMENT TASK	ILF	2 1 1	7+1+1 3+2+1 4	L L L
LOCALIZATION	EIF	1	2	L
Z111 Z113 Z115 Z213 Z311	EI	3 2 4 4 4	9+1+1 3+1+1 6+1+1 9+1+1 8+1+1	H A H H H
Z411	EO	3	<=19	A
Z421	EQ	4	8+1+1	H

STAGE 1

Type of constituent	Level of complexity		
	L	**A**	**H**
ILF	2*7	0*10	0*15
EIF	2*5	0*7	0*10
EI	1*3	0*4	4*6
EO	0*4	1*5	0*7
EQ	0*3	0*4	1*6
Suma	**27**	**5**	**30**

Using the formula for calculating *UFP* (1.d), one gets
UFP = 26+9+30 = 62

STAGE 2

Having fixed the level of input for every characteristics, the total degree of input *DI* (2.b)

$$DI = 21$$

Having inserted this value in the formula (2.c), the technical complexity fraction takes the following form:

$$TCF = 0.62 + 0.21 = 0.86$$

STAGE 3

Having put the calculated values of *UFP* and *TCF* into the formula for calculating *FP* (3) one gets

$$FP = 62*0.86 = 53,3$$

EXEMPLARY APPLICATION BIGNESS IN THE FUNCTIONAL POINTS IFPUG

- SAP – 296 764
- Ms Vista – 157 658
- Ms Office 2007 – 93 498
- Apple I Phone – 18 398
- Google Search Engine – 16 895
- Linux – 17 672

Correlation between lines of code (LOC) and functional points (FP)

Programming by	Average LOC/PF
• Asembler	320
• C	128
• Cobol	106
• Fortran	106
• Pascal	90
• PL/1	65
• C++	64
• Ada95	53
• Visual Basic	32
• Smalltalk	22
• Power Builder (code generator)	16
• SQL	12

SCALE OF THE SOFTWARE SIZE
[TAXONOMY ACCORDING TO SPR (SOFTWARE PRODUCTIVITY RESEARCH COMPANY]

Descriptive size of an application	FP	CLOC
Very small	1 - 100	1 - 10
Small	100 – 1 000	10 - 100
Below average	1 000 – 2 500	100 - 250
Average	2 500 – 10 000	250 – 1 000
Big system	10 000 – 25 000	1 000 – 2 500
Very big system	25 000 – 100 000	2 500 – 10 000
Huge system	> 100 000	> 10 000

Remark regarding the CLOC column:
types of programming have not been taken into account