Gordana Rakić

An update of the "COST ESTIMATION" topic

Objectives

- General differences between manual for function point analysis from 2005 and manual from 1985.
- Apply rules from 2005 to case study “Seminar organization”
- Make changes on slides for Topic 6

Topic 6
Cost Estimation

DAAD Project
"Joint Course on Software Engineering"

**Humboldt University Berlin, University of Novi Sad, University of Plovdiv,
University of Skopje, University of Belgrade, University of Niš, University of Kragujevac**

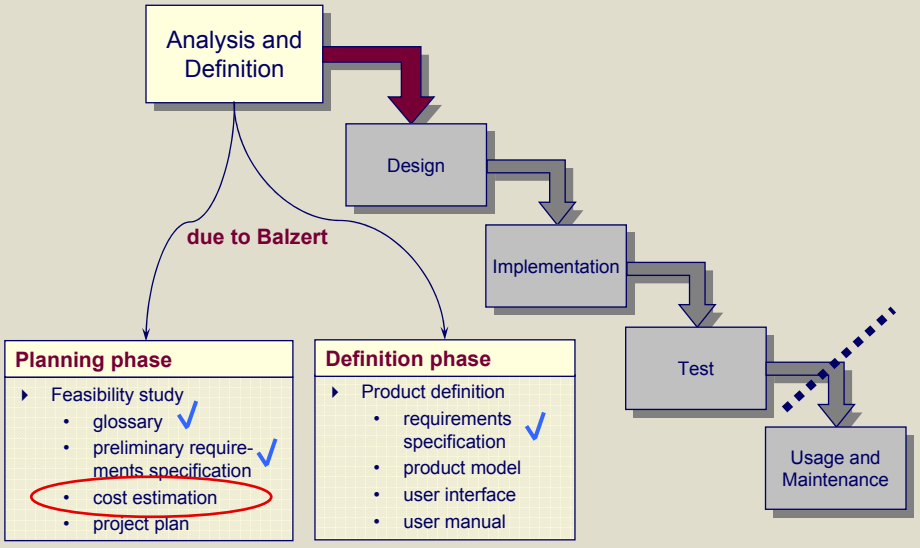
**Parts of this topic use material from the textbook
H. Balzert, "Software-Technik", Vol. 1, 2nd ed., Spektrum Akademischer Verlag, 2001**

Version: Nov. 10, 2004 (D Sep. 29, 2003)

6. Cost Estimation

- a) Cost measures and influencing factors for cost estimation
- b) Approaches (Basic methods)
- c) Function Point method
- d) Example of applying FP method

Classical waterfall model (1970)



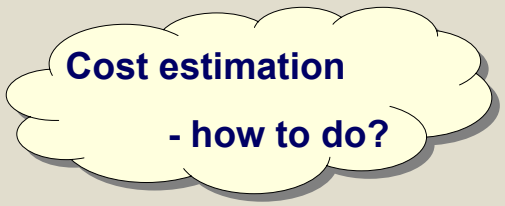
Motivation

Why should costs be estimated?

project planning:
dates, personnel

Contracts with
fixed prices

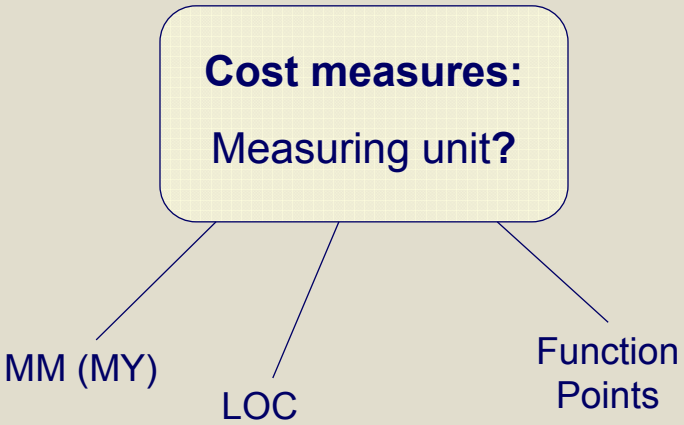
Initial ideas …?



Questions:

- On which basis should be estimated (Input)?
- What kind of result is expected (Output)?

Measuring units for cost estimation



Cost measures: MM, MY

- What time will the development take?
  - e.g. 10 years for one employee = 10 MY
    = 12*10 MM (man-month)
    = 5 years for 2 employees
    = 1 year for 10 employees
    = 1 day for … employees … ??

- but: communication effort raises with growing number of team members!

Optimal software development time

Optimal software development time (cf. Boehm 1981)

= 2,5 * (Effort in MM)^{S}

(e.g. S = 0,35 for online systems,
S = 0,32 for real time systems)

Example:

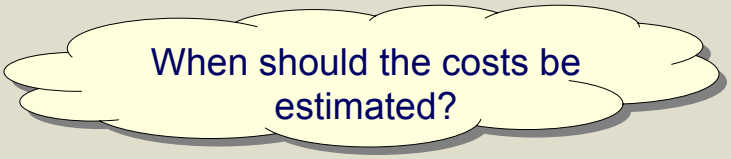
- Estimated development effort: 9 MM
- Optimal duration = 2,5 * $9^{0,35}$ = 5,3 month
- Number of employees
  = 9 MM / 5,3 month = 1,7 (i.e. 2)

Influencing factors for cost estimation

- What do the costs depend on?

  - Size of the problem to solve
    - length of program: LOC
  - Quality requirements
    - e.g. high efficiency, high safety…
  - Quality of developers
    - productivity
  - Company's experiences with similar tasks

Time of estimation



- *As early as possible*
- *At the beginning* of analysis and definition phase, based on the preliminary requirements specification
- *after* analysis and definition phase, based on the (final) requirements specification

6. Cost Estimation

a) Cost measures and influencing factors for cost estimation

b) Approaches (Basic methods)

c) Function Point method

d) Example of applying FP method

Basic approaches to cost estimation

- → Analogy method
- → Multiplicator method
- → Weighting method
- → Percentage method
- Relation method
- Parametric equations

Analogy method

Comparison of development to be estimated and already finished product developments based on similarity criteria.

- Criteria
  - same or similar application area
  - same or similar product size
  - same or similar degree of complexity
  - same programming language etc.

Example: analogy method

- Pascal compiler: 20 MM
- Modula2 compiler?
  - 20% new constructs
  - 50% of code reusable
  - 50% have to be reworked
- Estimation
  - 20% additional new development with high complexity:
    → 4 MM * 1,5 (because of complexity) = 6 MM
  - 50% slightly modified: 1/4 of 10 MM = 2,5 MM
  - 50% total rework: 10 MM
  - Estimation for Modula2 compiler: 18,5 MM

Multiplicator method

- Partitioning of system into subsystems
- Cost estimation for the subsystems
- Inclusion of weighting factors for complexity

Example: Multiplicator method

- Result of partitioning the software product for that the cost were to be estimated:

Category	sub-products	Sum LOC	Cost factor	LOC evaluated
Control logic	1*500 LOC	500	1,8	900
I/O logic	1*700+2*500	1700	1,5	2550
Data management	1*800+2*250	1300	1,0	1300
Algorithms	1*300+5*100	800	2,0	1600
Sum				6350

Weighting method

- Determine factors for estimation
  - e.g. Reusability is an important attribute, algorithms are little complex
- Compute with calculation formula
  (weighted factors)

→ Function Point method

Percentage method

- Analysis of earlier company projects:
  - Cost distribution between single phases
    (e.g. Analysis and Definition 30%, Design 30%, Implementation 15-20%, Test 20-25%)
    (Values from Bertelsmann)
- Cost of one phase must be known (e.g. A&D) or needs to be estimated with a different method
  → Extrapolation on whole project

Usage of cost estimation methods

Estimation method	year	basic method	usage	factors
EGW	77	W	P, D	1, 4
Boeing	77	W, Pe, M	P, D	1, 4
IFA-PASS	77	A, P	P	1, 2, 4
DOTY	77	W, Pe	P	1, 2, 4
GRIFFIN	77	W, P	P	1, 4
Schneider	78	Pe	P	1
INVAS	80	R, G	I	1, 2, 3, 4
ZKP	80	W, P	AD	1, 4
COCOMO	**81**	**W, Pe**	**P**	**1, 2, 3, 4**
Function Point	**81**	**A, W**	**I**	**1, 2, 3, 4**

Variants:
- Data Points 91
- Object Points 96

Source: Balzert, Vol.1, 2nd Ed., p. 79

Legend for cost estimation methods

Basic method:	A – Analogy method M – Multiplicator method R – Relation method W – Weighting method P – Percentage method Pe – Parametric equations
Time of application	P – Planning phase AD – Analysis and Definition phase D – Design phase I – Iterative process
Factors considered	1 – Quantity 2 – Quality 3 – duration of development 4 - Productivity

6. Cost Estimation

a) Cost measures and influencing factors for cost estimation

b) Approaches (Basic methods)

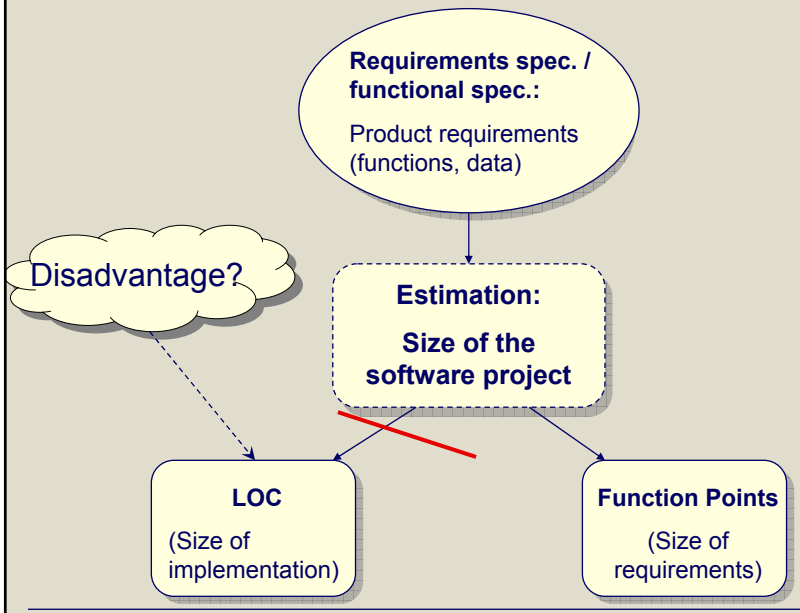
c) Function Point method

d) Example of applying FP method

History

- Allan J. Albrecht
  * 6.2.1927 in Pittston, PA., USA
  IBM Senior Engineer and Program Manager (retired)
- Inventor of the Function Point method used for cost estimation of software developments during his time at IBM (1979).

Function Point method: possible measuring units

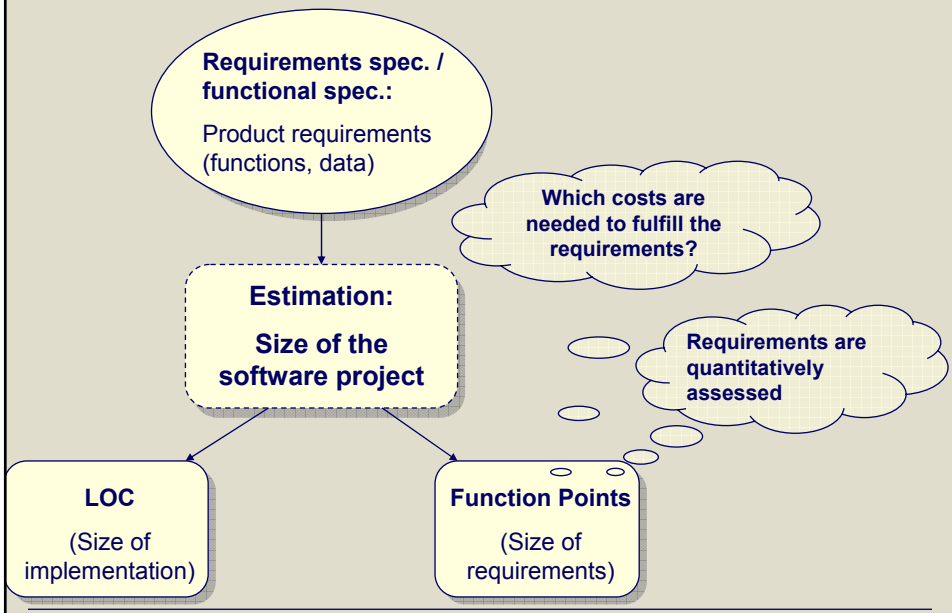


Relation between Function Points and LOC with different programming languages

Assembler	320
C	150
COBOL	106
FORTRAN	106
Pascal	91
PL/1	80
Ada	71
Prolog	64
APL	32
C++	29*
Smalltalk	21
Spreadsheet languages	6

Source: Yourdon, p.155; * Jones91

Interpretation of Function Points



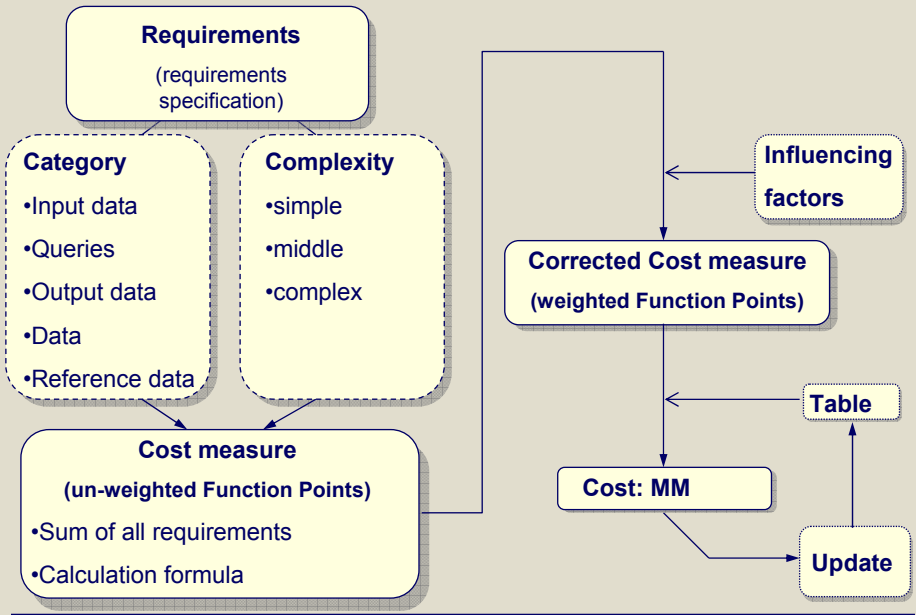
FP method: Characteristics

- relatively precise
  - Experiments: Deviation from actual value: 10%
- subjective
  - but: mostly independent of estimating person
  - average deviation between 2 persons: 12%
- Experiences needed for method application
- Industry standard in cost estimation methods
  - worldwide approx. 500 big companies, e.g. Boeing, IBM, VW
- Precondition: commercial applications
  - e.g. banks, insurances, management, warehousing
  - not: compilers and other system software!

FP Methode: Advantages and disadvantages

Advantages	Disadvantages
Based on product requirements	Limited to commercial applications
iterative method (requirements spec - functional spec - product model)	Tends to underestimate (incomplete requirements)
Estimation possible at early points of time	
Easy to learn	
Good estimation preciseness	
Tool support available	

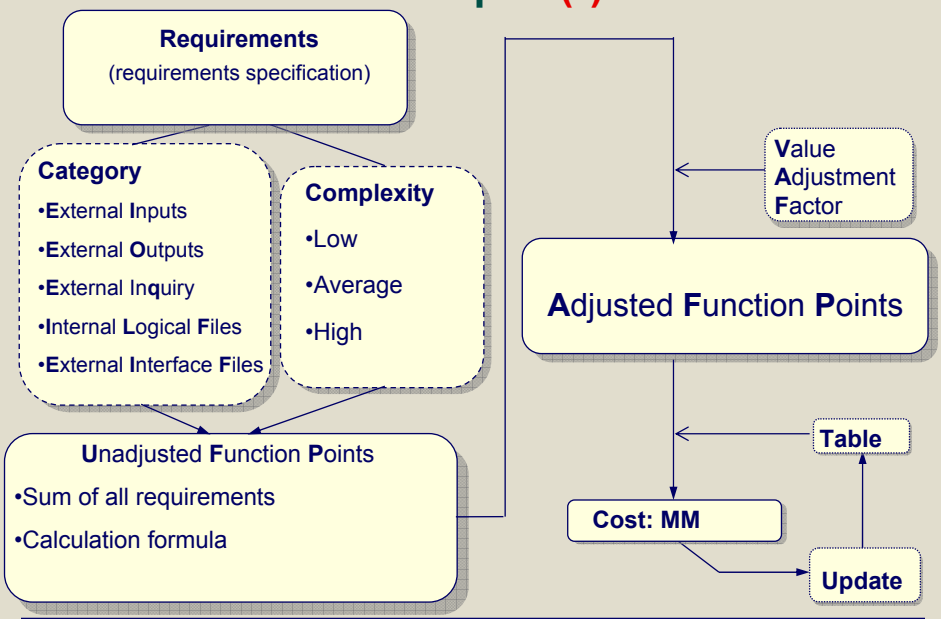
FP method: Principles



FP method: Counting process(*)

- **Determine type of function point count**
  - Development project function point count
  - Enhancement project function point count
  - Application function point count
- **Determine the application boundary**
- **Identify and rate transactional function types to determine their contribution to the unadjusted function point count**
- **Identify and rate data function types to determine their contribution to the unadjusted function point count**
- **Determine the value adjustment factor (VAF)**
- **Calculate the adjusted function point count**
  - equation depends on type of function point count

FP method for function types : Principles(*)



Calculation form IBM 1985

Category	Number	Classification	Weighting	Sums
Input data		simple	x 3	=
		middle	x 4	=
		complex	x 6	=
Queries		simple	x 3	=
		middle	x 4	=
		complex	x 6	=
Output data		simple	x 4	=
		middle	x 5	=
		complex	x 7	=
Data		simple	x 7	=
		middle	x 10	=
		complex	x 15	=
Reference data		simple	x 5	=
		middle	x 7	=
		complex	x 10	=
Sum			E1	=
Influencing factors (Function Point value can be changed by +/- 30%)		1 Integration with other applications (0-5)		=
		2 Decentralized data/ processing (0-5)		=
		3 Transaction rate (0-5)		=
		4 Processing logic a Arithmetic operations (0-10) b Control procedures (0-5) c Exception handling (0-10) d Logic (0-5)		= = = = =
		5 Reusability (0-5)		=
		6 Data stock conversions (0-5)		=
		7 Adaptability (0-5)		=
Sum of the 7 influences		E2		=
Evaluation of infl. factors =E2/100 + 0.7		E3 =		=
Weighted function Points: E1 * E3				=

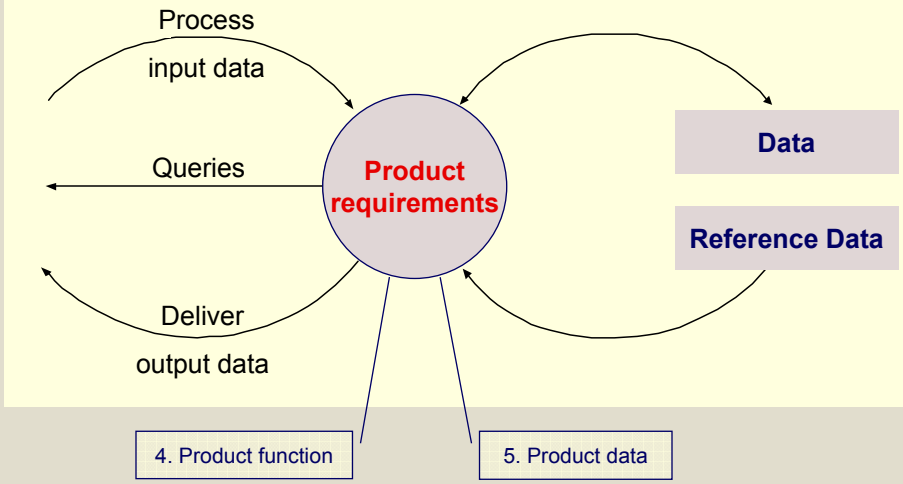
Source: IBM 85, p.12

Source: Balzert, Vol. 1, p. 84

Calculation form (*)

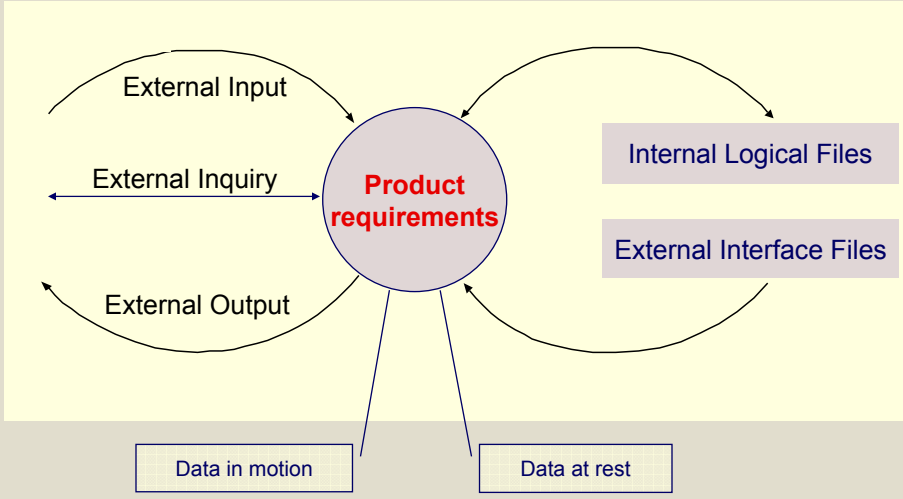
Category	Number	Classification	Weighting	Sums
External Input		Low	x 3	=
		Average	x 4	=
		High	x 6	=
External Inquiry		Low	x 3	=
		Average	x 4	=
		High	x 6	=
External Output		Low	x 4	=
		Average	x 5	=
		High	x 7	=
Internal Logical Files		Low	x 7	=
		Average	x 10	=
		High	x 15	=
External Interface Files		Low	x 5	=
		Average	x 7	=
		High	x 10	=
Sum			UFP	=
General System Characteristic		1 Data communications 2 Distributed data processing 3 Performance 4 Heavily used configuration 5 Transaction rate 6 On-Line data entry 7 End-user efficiency 8 On-Line update 9 Complex processing 10 Reusability 11 Installation ease 12 Operational ease 13 Multiple sites 14 Facilitate change		= = = = = = = = = = = = = =
Sum of the 14 influences		SUM(GSC)		=
Evaluation of VAF =SUM/100 + 0.65		VAF =		=
Adjusted function Points: UFP * VAF		AFP =		=

Requirements categories



Source: Balzert

Requirements categories(*)



Source: Balzert

Requirements categories: product functions

Product functions

Input data: function reads and processes input data

Output data: function internally computes output data

Query: function delivers output data via searching for information, e.g in databases (output lists)

Requirements categories: Data in motion (*)

Data in motion

External Input: is an elementary process in which data crosses the boundary from outside to inside

External Output: an elementary process in which ***derived data*** passes across the boundary from inside to outside

External Inquiry: an elementary process with both input and output components that result in data retrieval from one or more internal logical files and external interface files

Requirements categories: product data

Product data

Data: basic product data

Reference data: files with additional information (tables, read only files) e.g. to make links between logical data groups (secondary files)

Requirements categories: data at rest (*)

Data at rest

Internal Logical Files: a user identifiable group of logically related data that resides entirely within the application boundary and is maintained through External Inputs

External Interface Files: a user identifiable group of logically related data that is used for reference purposes only

Complexity of requirements: input data

Classifying *input data*

Criterion	simple	middle	complex
Number of different data elements	1-5	6-10	>10
Input correctness check	formal	formal logical	formal logical DB access
Expected user guidance	low	normal	high

DB = data base

Source: Balzert Vol.1 (1. edition) p. 80-82

Complexity of requirements: queries and output data

Classifying *queries*

Criterion	simple	middle	complex
Number of different keys	1	2	>2
Expected user guidance	low	normal	high

Classifying *output data*

Criterion	simple	middle	complex
Number of columns	1-6	7-15	>15
Number of different data elements	1-5	6-10	>10
Group change	1	2-3	>3
Data elements w/ print preprocessing	none	some	many

Source: Balzert Vol.1 (1. edition) p. 80-82

Complexity of requirements: data

Classifying *data*

Criterion	simple	middle	complex
Number of keys / record types	1	2	>2
Number of different data elements	1-20	21-40	>40
Data set available (no new architecture needed)	yes	-	no
Change of implemented data structure / data set	no	yes	-

Source: Balzert Vol.1 (1. edition) p. 80-82

Complexity of requirements: reference data

Classifying *reference data*

Criterion	simple	middle	complex
read only files			
Number of different data elements	1-5	6-10	>10
Number of keys / record types	1	2	>2
tables			
Number of different data elements	1-5	6-10	>10
Number of dimensions	1	2	3

Source: Balzert Vol.1 (1. edition) p. 80-82

Complexity of requirements: (*)
Data in motion - transactions

External Input

FTR \ DET	1 - 4	5 - 15	> 15
< 2	Low (3)	Low (3)	Average(4)
2	Low (3)	Average(4)	High(6)
> 2	Average(4)	High(6)	High(6)

unique user recognizable, non-recursive (non-repetitive) field

type referenced by a transaction (internal logical file or external interface file)

External Output

FTR \ DET	1 - 5	6 - 19	> 19
< 2	Low (4)	Low (4)	Average(5)
2, 3	Low (4)	Average(5)	High(7)
> 3	Average(5)	High(7)	High(7)

Source: Balzert Vol.1 (1. edition) p. 80-82

Complexity of requirements: (*)
Data in motion - transactions

External Inquiry

FTR \ DET	1 - 5	6 - 19	> 19
< 2	Low (3)	Low (3)	Average(4)
2, 3	Low (3)	Average(4)	High(6)
> 3	Average(4)	High(6)	High(6)

- DET - Data Element Type is a unique user recognizable, nonrecursive (non-repetitive) field
- FTR - File Type Referenced is a file type referenced by a transaction. An FTR must also be an internal logical file or external interface file.
- RET - Record Element Type is user recognizable sub group of data elements within an ILF or an EIF

Complexity of requirements: (*)
Data at rest - files

Internal Logical Files

RET \ DET	1 - 19	20 - 50	> 50
1	Low (7)	Low (7)	Average(10)
2, .., 5	Low (7)	Average(10)	High(15)
> 5	Average(10)	High(15)	High(15)

unique user recognizable, non-recursive (non-repetitive) field

user recognizable sub group of data elements within an ILF or an EIF

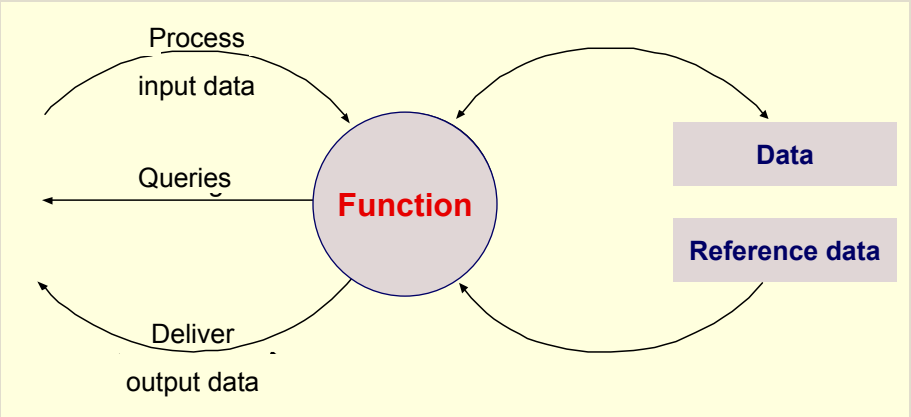
External Interface Files

RET \ DET	1 - 19	20 - 50	> 50
1	Low (5)	Low (5)	Average(7)
2, .., 5	Low (5)	Average(7)	High(10)
> 5	Average(7)	High(10)	High(10)

Source: Balzert Vol.1 (1. edition) p. 80-82

General Approach

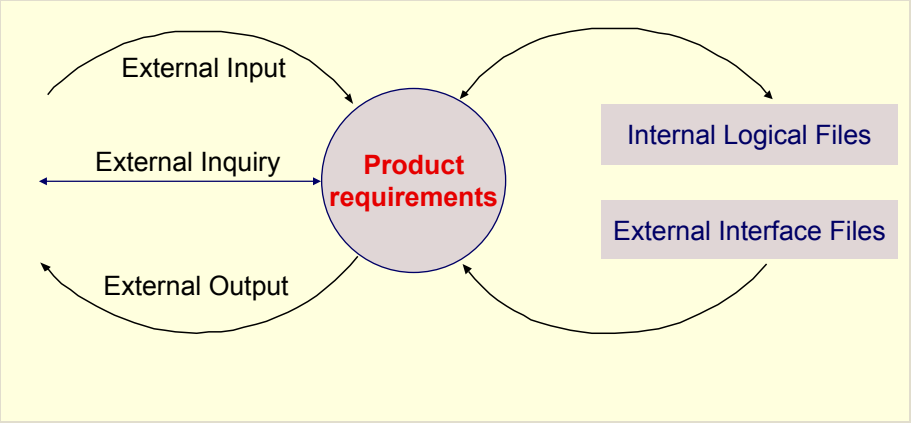
1. Step: Categorize each requirement



Source: Balzert

General Approach (*)

1. 1.Step: Categorize each requirement



Source: Balzert

1. 1. Step: Categorizing
(Preliminary requirements specification v3.0)

- /PF 70/: „Participant list per event“
  - This requirement belongs to the category *output data*.
- /PF 80/ „Participation certificate for each participant“
  - This requirement belongs to the category *output data*, too.

1. Step: Categorizing transaction and files(*)

(Preliminary requirements specification v3.0)

- /PF 70/: „Participant list per event“
  - This requirement belongs to the category *External Output*.
- /PF 80/ „Participation certificate for each participant“
  - This requirement belongs to the category *External Output*, too.

2. Step: Assigning complexity

- Put each requirement into one of the classes:
  - *simple*,
  - *middle* or
  - *complex*
  - Example:
    - Because there are no specifics concerning the outputs given in the preliminary requirements specification, all outputs are classified as being of *middle* complexity.

• 2. Step: Assigning complexity (*)

- Put each requirement into one of the classes:
  - *Low*,
  - *Average* or
  - *High*
  - Example:
    - Because there are no specifics concerning the outputs given in the preliminary requirements specification, all outputs are classified as being of *Average* complexity.

3. Step: Filling out calculation form

- Requirement /PF 70/ contains *1* middle output, so does /PF 80/.
- Altogether there are 5 middle and 4 complex outputs.
  - Put a *5* into the number column of the second row belonging to outputs, and a *4* into the third row.
  - These numbers are multiplied by *5* resp. *7* (weighting factors).
  - This results in *25* resp. *28* Function Points.

- **3. Step: Filling out calculation form (*)**

- Requirement /PF 70/ contains *1 Average External Output*, so does /PF 80/.
- Altogether there are *5 Low* and *4 High External Outputs*.
  - Put a *5* into the number column of the second row belonging to External Outputs, and a *4* into the third row.
  - These numbers are multiplied by *5* resp. *7* .
  - This results in *25* resp. *28* Unadjusted Function Points.

Category	Number	Classification	Weighting	Sums
Input data		simple	x 3	=
		middle	x 4	=
		complex	x 6	=
Queries		simple	x 3	=
		middle	x 4	=
		complex	x 6	=
Output data		simple	x 4	=
		middle	x 5	=
		complex	x 7	=
Data		simple	x 7	=
		middle	x 10	=
		complex	x 15	=
Reference data		simple	x 5	=
		middle	x 7	=
		complex	x 10	=
Sum			E1	=
Influencing factors (Function Point value can be changed by +/- 30%)		1 Integration with other applications (0-5)		=
		2 Decentralized data/ processing (0-5)		=
		3 Transaction rate (0-5)		=
		4 Processing logic a Arithmetic operations (0-10) b Control procedures (0-5) c Exception handling (0-10) d Logic (0-5)		= = = = =
		5 Reusability (0-5)		=
		6 Data stock conversions (0-5)		=
		7 Adaptability (0-5)		=
Sum of the 7 influences		E2		=
Evaluation of infl. factors =E2/100 + 0.7		E3 =		=
Weighted function Points: E1 * E3				=

Source: IBM 85, p.12

Calculation form

Example: Values from preliminary requirements specification

Category	Number	Classification	Weighting	Sums
External Input		Low	x 3	=
		Average	x 4	=
		High	x 6	=
External Inquiry		Low	x 3	=
		Average	x 4	=
		High	x 6	=
External Output		Low	x 4	=
		Average	x 5	=
		High	x 7	=
Internal Logical Files		Low	x 7	=
		Average	x 10	=
		High	x 15	=
External Interface Files		Low	x 5	=
		Average	x 7	=
		High	x 10	=
Sum			UFP	=
General System Characteristic		1 Data communications 2 Distributed data processing 3 Performance 4 Heavily used configuration 5 Transaction rate 6 On-Line data entry 7 End-user efficiency 8 On-Line update 9 Complex processing 10 Reusability 11 Installation ease 12 Operational ease 13 Multiple sites 14 Facilitate change		= = = = = = = = = = = = = =
Sum of the 14 influences		SUM(GSC)		=
Evaluation of VAF =SUM/100 + 0.65		VAF =		=
Adjusted function Points: UFP * VAF		AFP =		=

Source: IBM 85, p.12

Calculation form

Example: Values from preliminary requirements specification (*)

Category	Number	Classification	Weighting	Sums
Input data	0	simple	x 3	= 0
	11	middle	x 4	= 44
	4	complex	x 6	= 24
Queries	0	simple	x 3	= 0
	0	middle	x 4	= 0
	0	complex	x 6	= 0
Output data	0	simple	x 4	= 0
	5	middle	x 5	= 25
	4	complex	x 7	= 28
Data	6	simple	x 7	= 42
	0	middle	x 10	= 0
	0	complex	x 15	= 0
Reference data	0	simple	x 5	= 0
	0	middle	x 7	= 0
	0	complex	x 10	= 0
Sum			E1	= 163

un-weighted function points

(*)

Category	Number	Classification	Weighting	Sums
External Input	0	Low	x 3	= 0
	11	Average	x 4	= 44
	4	High	x 6	= 24
External Inquiry	0	Low	x 3	= 0
	0	Average	x 4	= 0
	0	High	x 6	= 0
External Output	0	Low	x 4	= 0
	5	Average	x 5	= 25
	4	High	x 7	= 28
Internal Logical Files	6	Low	x 7	= 42
	0	Average	x 10	= 0
	0	High	x 15	= 0
External Interface Files	0	Low	x 5	= 0
	0	Average	x 7	= 0
	0	High	x 10	= 0
Sum			UFP	= 163

Unadjusted Function Points

4. and 5. Step

- 4. Step:
  - Evaluating requirements considering the influencing factors
- 5. Step:
  - Calculating *weighted* Function Points

Influencing factors (Function Point value can be changed by +/- 30%)	1 Integration with other applications (0-5)	= 0
	2 Decentralized data/ processing (0-5)	= 0
	3 Transaction rate (0-5)	= 3
	4 Processing logic a Arithmetic operations (0-10) b Control procedures (0-5) c Exception handling (0-10) d Logic (0-5)	 = 3 = 3 = 3 = 3
	5 Reusability (0-5)	= 0
	6 Data stock conversions (0-5)	= 0
	7 Adaptability (0-5)	= 3
Sum of the 7 influences	E2	= 18
Evaluation of infl. factors =E2/100 + 0.7	E3 = 18 / 100 + 0,7	= 0,88
Weighted function Points: E1 * E3		= 143

4. and 5. Step (*)

- 4. Step:
  - Evaluating requirements considering the 14 general system characteristics
- 5. Step:
  - Calculating *Adjusted Function Points - equation depends on* type of function point count (development project function point count, enhancement project function point count, application function point count)

General system characteristics(*)

- **Data communications** - how many communication facilities are there to aid in the transfer or exchange of information with the application or system
- **Distributed data processing** - how are distributed data and processing functions
- **Performance -** did the user require response time or throughput
- **Heavily used configuration** - how heavily used is the current hardware platform where the application will be executed
- **Transaction rate** - how frequently are transactions executed daily, weekly, monthly, etc

General system characteristics(*)

- **On-Line data entry** - what percentage of the information is entered On-Line
- **End-user efficiency -** was the application designed for end-user efficiency
- **On-Line update** - how many ILF’s are updated by On-Line transactions
- **Complex processing** - does the application have extensive logical or mathematical processing
- **Reusability** - was the application developed to meet one or many user’s needs

General system characteristics(*)

- **Installation ease** - how difficult is conversion and installation
- **Operational ease -** how effective and/or automated are start-up, backup, and recovery procedures
- **Multiple sites** - was the application specifically designed, developed, and supported to be installed at multiple sites for multiple organizations
- **Facilitate change -** was the application specifically designed, developed, and supported to facilitate change

(*)

General System Characteristic	1	Data communications	= 0
	2	Distributed data processing	= 0
	3	Performance	= 0
	4	Heavily used configuration	= 0
	5	Transaction rate	= 3
	6	On-Line data entry	= 0
	7	End-user efficiency	= 0
	8	On-Line update	= 0
	9	Complex processing	= 4
	10	Reusability	= 0
	11	Installation ease	= 0
	12	Operational ease	= 0
	13	Multiple sites	= 0
	14	Facilitate change	= 3
Sum of the 14 values	SUM		= 10
Evaluation of VAF =SUM/100 + 0.65	VAF = 10 / 100 + 0,65		= 0.75
Adjusted function Points: UFP * VAF	AFP = 163 * 0.75		= 122.25

6. and 7. Step

- 6. Step:
  - Looking up costs in MM

    (Precondition: empirical determination of relation between FP and MM)
- 7. Step:
  - Updating empirical data

Function Point value pairs (1986/1991)

Function P.	IBM MM	VW MM	Function P.	IBM MM	VW MM	Function P.	IBM MM
50	2,3		1200	145,2	207,8	3300	547
100	5,6		1300	161,3	237,8	3400	568,8
150	9,5		1400	177,7	273,2	3500	590,8
200	13,9	11,7	1500	194,6	319,1	3600	613,1
250	18,6	19,3	1600	211,7		3700	635,5
300	23,6	27,1	1700	229,3		3800	658,1
350	28,9	35	1800	247,1		3900	680,9
400	34,4	43	1900	265,3		4000	703,9
450	40,1	51,1	2000	283,7		4100	727
500	46,1	59,6	2100	302,4		4200	750,4
550	52,2	68,2	2200	321,5		4300	773,9
600	58,5	77	2300	340,7		4400	797,5
650	65	86,1	2400	360,3		4500	821,4
700	71,6	95,3	2500	380,1		4600	845,4
750	78,4	104,8	2600	400,1		4700	869,6
800	85,3	114,6	2700	420,4		4800	893,9
850	92,4	124,7	2800	441		4900	918,4
900	99,6	135,2	2900	461,7		5000	943,1
950	106,9	146	3000	482,7		5100	967,9
1000	114,4	157,3	3100	503,9		5200	992,8
1100	129,6	181,3	3200	525,4			

Source: Noth, Kretzschmar 86 / Knöll, Busse 91 (Balzert, Vol.1, 1. edition)

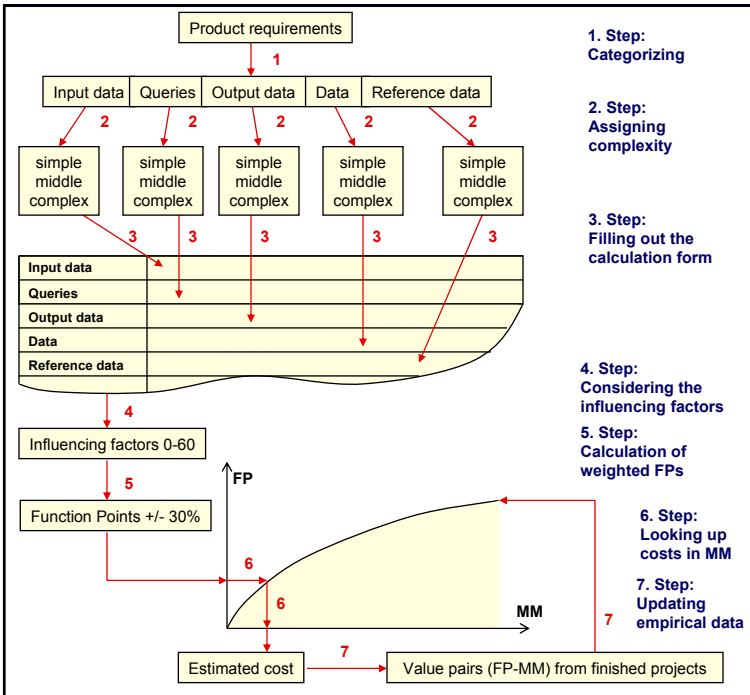
→ costs are growing *non-linear*

Function Point value pairs (2000)

FP	IBM MM	FP	IBM MM	FP	IBM MM
50	5	700	52	1700	142
100	8	750	56	1800	153
150	11	800	60	1900	164
200	14	850	64	2000	175
250	17	900	68	2100	188
300	20	950	72	2200	201
350	24	1000	76	2300	215
400	28	1100	85	2400	230
450	32	1200	94	2500	245
500	36	1300	103	2600	263
550	40	1400	112	2700	284
600	44	1500	122	2800	307
650	48	1600	132	2900	341

Source: Balzert, Vol.1, 2. edition

Procedure (overview of steps)



1. Step: Categorizing

2. Step: Assigning complexity

3. Step: Filling out the calculation form

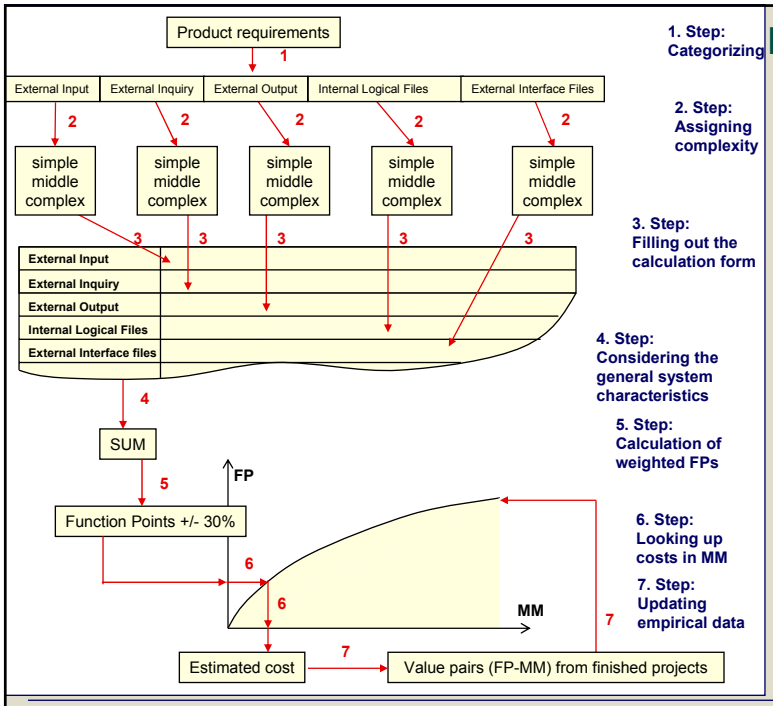
4. Step: Considering the influencing factors

5. Step: Calculation of weighted FPs

6. Step: Looking up costs in MM

7. Step: Updating empirical data

Procedure (overview of steps) (*)



6. Cost Estimation

a) Cost measures and influencing factors for cost estimation

b) Approaches (Basic methods)

c) Function Point method

d) Example of applying FP method

Example: Preliminary Requirements Specification „Seminar organization“ V 2.3 (1)

- /PF 10/ Adding new, changing and deleting customers’ data (participants, prospects).

/PF10/:

These are three separate inputs (Adding, Changing, Deleting).

Adding a new client is certainly most extensive, there are probably more than 10 data elements to be gathered, a logic input correctness check is needed (consistency check: zip-code/place), a writing access to the data base is needed, user guidance is expected to be high (automatic positioning of cursor, field centered editing). Adding a new client is a complex input.

Repetition: Classifying requirements (complexity)

Classifying *input data*

Criterion	simple	middle	complex
Number of different data elements	1-5	6-10	>10
Input correctness check	formal	formal logical	formal logical DB access
Expected user guidance	low	normal	high

DB = data base

Source: Balzert Vol.1 (1. edition) p. 80-82

Example: Preliminary Requirements Specification „Seminar organization“ V 2.3 (1)(*)

- /PF 10/ Adding new, changing and deleting customers’ data (participants, prospects).

/PF10/:

These are three separate inputs (Adding, Changing, Deleting).

Adding a new client is certainly most extensive, there are probably more than 15 data elements to be gathered, and 2 or 3 file types referenced (file for client, a logic input correctness check is needed - consistency check: zip-code/place, and file for company if a client is company’s associate).

Adding a new client is a high external input.

Repetition: Classifying requirements (complexity)(*)

External Input

FTR \ DET	1 - 4	5 - 15	> 15
< 2	Low (3)	Low (3)	Average(4)
2	Low (3)	Average(4)	High(6)
> 2	Average(4)	High(6)	High(6)

Example: Requirements Specification „Seminar organization“ V 2.3 (2)

/PF10/ contd.:

During a change of client's data the data base is read from and written on. User guidance is needed to be usual, the number of changed data elements may vary from small to high. Therefore classfying this input as middle seems sufficient.

Deleting a client's entry demands logical checks and a data base access on seminar bookings /PF50/. Deleting is therefore also classified as being of middle complexity.

Result: *1* complex input, *2* middle inputs

Example: Requirements Specification „Seminar organization“ V 2.3 (2)(*)

/PF10/ contd.:

During a change of client's data 1, 2, or 3 file types referenced may be used and the number of changed data elements may vary from small to high. Therefore classifying this input as average seems sufficient.

Deleting a client's entry demands logical checks and access on seminar bookings file /PF50/. Hence, there are 2 file types referenced, and less then 5 data elements. Deleting is therefore also classified as being of low complexity.

Result: *1* high external input, *1 average external input, 1* low external input

Example: Requirements Specification „Seminar organization“ V 2.3 (3)

- /PF 20/ Information of customers (registration affirmation, checkout affirmation, change information, invoice, advertising)

/PF20/

These are *five* separate *outputs*.

Because there are *no specifics in the requirements specification* available and most of these outputs are combinations of a few data elements with some data and standard texts, they are classified as being of middle complexity.

Result: *5* middle outputs

Example: Requirements Specification „Seminar organization“ V 2.3 (3)(*)

- /PF 20/ Information of customers (registration affirmation, checkout affirmation, change information, invoice, advertising)

/PF20/

These are *five* separate *outputs*.

Because there are *no specifics in the requirements specification* available and most of these outputs are combinations of a few data elements with some data and standard texts, they are classified as being of average complexity.

Result: *5* average external outputs

Example: Requirements Specification „Seminar organization“ V 2.3 (4)

/PF30/

As in /PF10/, but respectively for seminar events and seminar types.

Result: *2* complex and *4* middle inputs

/PF40/

As in /PF10/

Result: *1* complex and *2* middle inputs

/PF50/

To book a presentation it is only necessary to link the customer with the corresponding seminar event. So there are only a few data elements involved, however a logical check with data base access is needed. These 3 inputs are classified as being of middle complexity.

Result: *3* middle inputs

Example: Requirements Specification „Seminar organization“ V 2.3 (4)(*)

/PF30/ There are 6 external inputs

Seminar type

Adding: DET > 15, FTR = 1, so this is average external input

Changing: number of changed data elements may vary from small to high (5 < DET < 15), FTR = 1, so this is low external input

Deleting: DET < 5, FTR = 2 (seminar types, seminar presentation), so this is low external input

Seminar presentation

Adding: DET > 15, FTR > = 2, so this is high external input

Changing: number of changed data elements and file type referenced may vary from small to high (5 < DET < 15, 1 <= FTR <= 3), so this is average external input

Deleting: DET < 5, FTR = 2 (seminar presentation, seminar booking), so this is low external input

Result: 3 low external inputs, 2 average external inputs, 1 high external input

Example: Requirements Specification „Seminar organization“ V 2.3 (4)(*)

/PF40/

Number of data elements is >5 (for deletion <5), and number of file types referenced is >2 (lecturer, seminar type, seminar presentation,…)

Result: *2* high external inputs, *1* average external inputs

/PF50/

To book a presentation it is only necessary to link the customer with the corresponding seminar event, so there are 2 file types referenced. So there are only a few data elements involved (<= 5)

These 3 external inputs are classified as being of low complexity.

Result: *3* low external inputs

Example: Requirements Specification „Seminar organization“ V 2.3 (5)

/PF60/

An invoice has to contain data on the customer, the seminar event and the seminar type. This requires some data base accesses. The output will probably contain more than 10 data elements. This leads to a complex output.

Result: *1* complex output

/PF70/

As in /PF60/ these are three complex outputs.

Result: *3* complex outputs

Example: Requirements Specification „Seminar organization“ V 2.3 (5)(*)

/PF60/

An invoice has to contain data on the customer, the seminar event and the seminar type (3 file types referenced). The output will probably contain more than 6, but less then 19 data elements. This leads to a average external output.

Result: *1* average external output

/PF70/

As in /PF60/ these are three average outputs.

Result: *3* average external outputs

Function-Points am Beispiel: Lastenheft „Seminarorganisation“ V 2.3 (6)

/PF80/

Queries similar to the following should be answered:

When will the next seminar X take place?

Which company Y‘s associates participated the seminar X?

These are queries with *end user languages*. They do not count.

Example: Requirements Specification „Seminar organization“ V 2.3 (7)

Product data

/PD10/

This should be one *simple data stock* (1 key, number of different data elements < 20).

Result: *1* simple data stock

/PD20/

As in /PD10/ this is one *simple data stock*.

Result: *1* simple data stock

/PD30/

As in /PD10/, respectively for seminar event, seminar type and lecturers.

Result: *3* simple data stocks

/PD40/

As in /PD10/.

Result: *1* simple data stock

Example: Requirements Specification „Seminar organization“ V 2.3 (7)(*)

Product data

/PD10/

This should be one *low internal logical file* (number of different data elements - DET< 20, record element types < 5 (1 or 2)).

Result: *1* low internal logical file

/PD20/

As in /PD10/ this is one *low internal logical file* .

Result: *1* low internal logical file

/PD30/

As in /PD10/, respectively for seminar event, seminar type and lecturers.

Result: *3* low internal logical file

/PD40/

As in /PD10/.

Result: *1* low internal logical file

Un-weighted Function Points

Input data:	11	x middle (4)	=	44
	4	x complex (6)	=	24
Output data:	5	x middle (5)	=	25
	4	x complex (7)	=	28
Data:	6	x simple (7)	=	42
Function Points sum (E1)				163

Unadjusted Function Points(*)

External Inputs:	7 x Low (3)	=	21
	4 x Average(4)	=	16
	4 x High(6)	=	24
External Outputs:	9 x Average (5)	=	45
Internal Logical Files:	6 x Low (7)	=	42
Function Points sum (UFP):			148

Influencing factors

The influencing factors are considered as follows:

1.	*Integration* with other applications (0-5):	0
2.	*Decentralized data* / processing (0-5):	0
3.	*Transaction rate* (0-5) : *because of /PF10/: efficient DB access*	3
4.	*Processing logic*	
	a) Arithmetic operations (0-10): *more complex algorithms*	3
	b) Control procedures (0-5):	3
	c) Exception handling (0-10): *special cases*	3
	d) Logic (0-5):	3
5.	*Reusability* (0-5):	0
6.	*Data stock conversions* (0-5):	0
7.	*Adaptability* (0-5):	3
Sum of the seven influences: E2:		18

General system characteristics(*)

1 Data communications	0
2 Distributed data processing	0
3 Performance	0
4 Heavily used configuration	0
5 Transaction rate	3
6 On-Line data entry	2
7 End-user efficiency	1
8 On-Line update	1
9 Complex processing	4
10 Reusability	0
11 Installation ease	0
12 Operational ease	0
13 Multiple sites	0
14 Facilitate change	3
Sum of the 14 values	14

Finish: weighted FPs, MM, number of employees

- Evaluation of influencing factors: E3:
  - E2/100+0,7=18/100+0,7=0,88
- Weighted Function Points:
  - E1 * E3 = 163 * 0,88 = 143 FP
- Costs according to IBM table (interpolated): ≈ 8,5 MM
- Optimal development time
  - = 2,5 *8,5 ^ 0,35 [months] = 5,2 [months]

Average size of development team is:

Number of employees = 8,5 MM / 5,2 months = 1,6 ≈ 2 employees

Finish: Adjusted FPs, MM, number of employees(*)

standard function point equation

- Evaluation of in Value Adjustment Factor factors: VAF = SUM(GSC)/100+0,65=14/100+0,65=0,79
- Adjusted Function Points:
  - UFP * VAF = 148 * 0,79 = 116.92 FP
- Costs according to IBM table (table 1986/1991) (interpolated): ≈ 7 MM
- Optimal development time
  - = 2,5 *7 ^ 0,35 [months] = 4,94 [months]

Average size of development team is:

Number of employees = 7 MM / 4,94 months = 1.42 ≈ 1 employees

Category	Number	Classification	Weighting	Sums
Input data	0	simple	x 3	= 0
	11	middle	x 4	= 44
	4	complex	x 6	= 24
Queries	0	simple	x 3	= 0
	0	middle	x 4	= 0
	0	complex	x 6	= 0
Output data	0	simple	x 4	= 0
	5	middle	x 5	= 25
	4	complex	x 7	= 28
Data	6	simple	x 7	= 42
	0	middle	x 10	= 0
	0	complex	x 15	= 0
Reference data	0	simple	x 5	= 0
	0	middle	x 7	= 0
	0	complex	x 10	= 0
Sum			E1	= 163

(*)

Category	Number	Classification	Weighting	Sums
External Input	7	Low	x 3	= 21
	4	Average	x 4	= 16
	4	High	x 6	= 24
External Inquiry	0	Low	x 3	= 0
	0	Average	x 4	= 0
	0	High	x 6	= 0
External Output	0	Low	x 4	= 0
	9	Average	x 5	= 45
	0	High	x 7	= 0
Internal Logical Files	6	Low	x 7	= 42
	0	Average	x 10	= 0
	0	High	x 15	= 0
External Interface Files	0	Low	x 5	= 0
	0	Average	x 7	= 0
	0	High	x 10	= 0
Sum			UFP	= 148

Influencing factors (Function Point value can be changed by +/- 30%)	1 Integration with other applications (0-5)	= 0
	2 Decentralized data/ processing (0-5)	= 0
	3 Transaction rate (0-5)	= 3
	4 Processing logic a Arithmetic operations (0-10) b Control procedures (0-5) c Exception handling (0-10) d Logic (0-5)	 = 3 = 3 = 3 = 3
	5 Reusability (0-5)	= 0
	6 Data stock conversions (0-5)	= 0
	7 Adaptability (0-5)	= 3
Sum of the 7 influences	E2	= 18
Evaluation of infl. factors =E2/100 + 0.7	E3 = 18 / 100 + 0,7	= 0,88
Weighted function Points: E1 * E3		= 143

(*)

General System Characteristic	1	Data communications	= 0
	2	Distributed data processing	= 0
	3	Performance	= 0
	4	Heavily used configuration	= 0
	5	Transaction rate	= 3
	6	On-Line data entry	= 2
	7	End-user efficiency	= 1
	8	On-Line update	= 1
	9	Complex processing	= 4
	10	Reusability	= 0
	11	Installation ease	= 0
	12	Operational ease	= 0
	13	Multiple sites	= 0
	14	Facilitate change	= 3
Sum of the 14 values	SUM		= 14
Evaluation of VAF =SUM/100 + 0.65	VAF = 14 / 100 + 0,65		= 0.79
Adjusted function Points: UFP * VAF	AFP = 148 * 0.79		= 116.92

Second example:
preliminary requirements specification V 3.0

New: identify elementary functions as part of use cases

→ *for each use case, find the elementary functions contained in it*

Example: /PF10/ Informing: A customer asks for information about seminares or a mailing of a seminar catalogue.

- elementary functions: „give information “ and „send catalogue“

Further developments

- *Data Points* /Sneed 91/
- *Object Points* /Sneed 96/
- Extrapolation method /Sneed 99/

- *IFPUG*: International Function Point Users Group
  - http://www.ifpug.org

		1985			2005		
		Category and complexity	FP	note	Category and complexity	FP	note
PF10	Add	Compex input	6	Number of different data elements > 10, DB access, Expected user guidance - high	EI High	6	DET>15 2<=FTR<=3
	Change	Middle input	4	Number of different data elements- vary, DB access, Expected user guidance normal	EI Average	4	DET and FTR vary
	Delete	Middle input	4	DB access, logical checks	EI Low	3	DET<5 FTR=2

PF20	Print	Middle output	5	there are no specifics in the requirements specification	EO Average	5	there are no specifics in the requirements specification
	Print	Middle output	5	there are no specifics in the requirements specification	EO Average	5	there are no specifics in the requirements specification
	Print	Middle output	5	there are no specifics in the requirements specification	EO Average	5	there are no specifics in the requirements specification
	Print	Middle output	5	there are no specifics in the requirements specification	EO Average	5	there are no specifics in the requirements specification
PF30	Add	Compex input	6	As in /PF10/	EI Average	4	DET > 15, FTR = 1
	Change	Middle input	4	As in /PF10/	EI Low	3	DET vary FTR = 1
	Delete	Middle input	4	As in /PF10/	EI Low	3	DET < 5, FTR = 2
	Add	Compex input	6	As in /PF10/	EI High	6	DET > 15, FTR > = 2
	Change	Middle input	4	As in /PF10/	EI Average	4	DET and FTR vary
	Delete	Middle input	4	As in /PF10/	EI Low	3	DET < 5, FTR = 2

PF40	Add	Compex input	6	As in /PF10'	EI High	6	DET > 5, FTR> 2
	Change	Middle input	4	As in /PF10'	EI High	6	DET > 5, FTR > 2
	Delete	Middle input	4	As in /PF10'	EI Average	4	DET < 5, FTR > 2
PF50	Add	Middle input	4	DB access and logical checks few data elements	EI Low	3	DET < 5, FTR = 2
	Change	Middle input	4	DB access and logical checks few data elements	EI Low	3	DET<5 FTR=2
	Delete	Middle input	4	DB access and logical checks few data elements	EI Low	3	DET<5 FTR=2
PF60	Print	Complex output	7	Number of different data elements> 10, DB access	EO Average	5	6<=DET<=19 FTR=3
PF70	Print	Complex output	7	Number of different data element > 10, DB access	EO Average	5	6<=DET<=19 FTR=3
	Print	Complex output	7	Number of different data elements> 10, DB access	EO Average	5	6<=DET<=19 FTR=3

	Print	Complex output	7	Number of different data elements > 10, DB access	EO Average	5	6<=DET<=19 FTR=3
PF80	Queries		0	These are queries with end user languages. They do not count.		0	These are queries with end user languages. They do not count.
PD10	Data	Simple data stock	7	Number of different data elements< 20, 1 key	ILF Low	7	DET < 20 RET < 5
PD20	Data	Simple data stock	7	As in /PD10'	ILF Low	7	As in /PD10'
PD30	Data	Simple data stock	7	As in /PD10'	ILF Low	7	As in /PD10'
	Data	Simple data stock	7	As in /PD10'	ILF Low	7	As in /PD10'
	Data	Simple data stock	7	As in /PD10'	ILF Low	7	As in /PD10'
	Data	Simple data stock	7	As in /PD10'	ILF Low	7	As in /PD10'
PD40	Data	Simple data stock	7	As in /PD10'	ILF Low	7	As in /PD10'

2005			1985		
characteristic	**rang**	**value**	**characteristic**	**range**	**value**
Data communications	[0,..,5]	0			
Distributed data processing	[0,..,5]	0			
Performance	[0,..,5]	0			
Heavily used configuration	[0,..,5]	0			
Transaction rate	[0,..,5]	3	Transaction rate	[0,..,5]	3
On-Line data entry	[0,..,5]	2			
End-user efficiency	[0,..,5]	1			
On-Line update	[0,..,5]	1			
Complex processing	[0,..,5]	4	Processing logic		
			Arithmetic operations	[0,..,10]	3
			Control procedures	[0,..,5]	3
			Exception handling	[0,..,10]	3
			Logic	[0,..,5]	3

Reusability	[0,..,5]	0	Reusability	[0,..,5]	0
Installation ease	[0,..,5]	0			
Operational ease	[0,..,5]	0			
Multiple sites	[0,..,5]	0			
Facilitate change	[0,..,5]	3	Adaptability	[0,..,5]	3
			Data stock conversions	[0,..,5]	0
			Integration with other applications	[0,..,5]	0
			Decentralized data / processing	[0,..,5]	0