



Function Point Sizing

Innovative Solutions in Cost Estimating
Dayton SCEA Chapter November 4, 2002

Bruce Fad, PRICE Systems LLC,
bruce.fad@pricesystems.com



FASTER DECISIONS. BETTER DECISIONS.™



Definition of Function Points

“A software size measure. They measure the amount of information processing functionality contained within a software product. They are derived early in the software life cycle from requirements or design specifications, and are applied across diverse application domains and technology platforms.”

Practical Software Measurement, UWC 1997



SLOC vs. Function Point Comparison

<i>Measure</i>	<i>Advantages</i>	<i>Disadvantages</i>
Function Points	Technology independent Logical user view International acceptance (IFPUG)	Requires extensive training
Source Lines of Code (SLOC)	Can be done from physical implementation	Technology dependent Inconsistent rules Counter-indicator of productivity Unavailable until coded



Origin of Function Points

1979

- Allan J. Albrecht of IBM presented his published paper “Measuring Application Development Productivity” at the Joint SHARE/GUIDE/IBM Application Development Symposium
 - Output metric of software project should be valid for all languages
 - Should represent topics of concern to software users
 - Should be a measure of functionality



Function Point Software Sizing History

1983

Albrecht and John E. Gaffney, Jr. refined function point technique by expanding function type definitions and counting procedures

1986

International Function Point Users Group (IFPUG) organized

1993

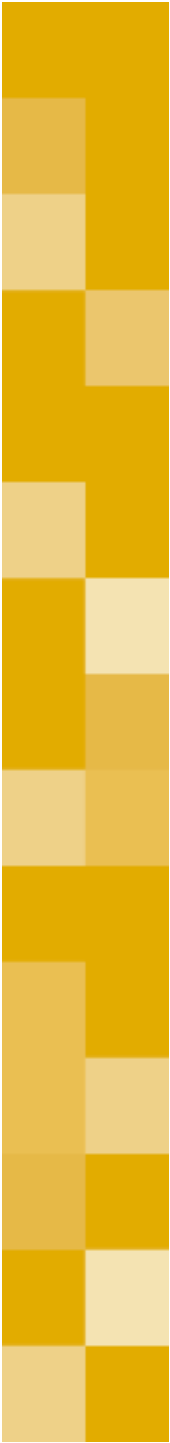
Certification for FP Specialists

1994

CPM 4.0, Guidelines to Software Measurement 1.0

Post 1994

CPM 4.1



In the eyes of the user, the visible external aspects of a software application are.....

- 1. Inputs***
- 2. Outputs***
- 3. Inquiries***
- 4. Internal Files***
- 5. Interfaces***

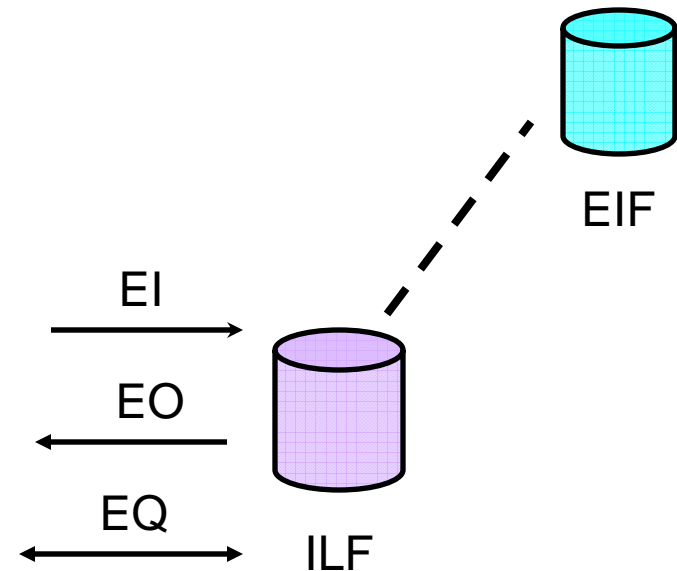
Components of Function Points

Data Functions:

- Internal Logical File (ILF)
- External Interface File (EIF)

Transactional Functions:

- External Input (EI)
- External Output (EO)
- External Query (EQ)



Source: IFPUG CPM 4.0

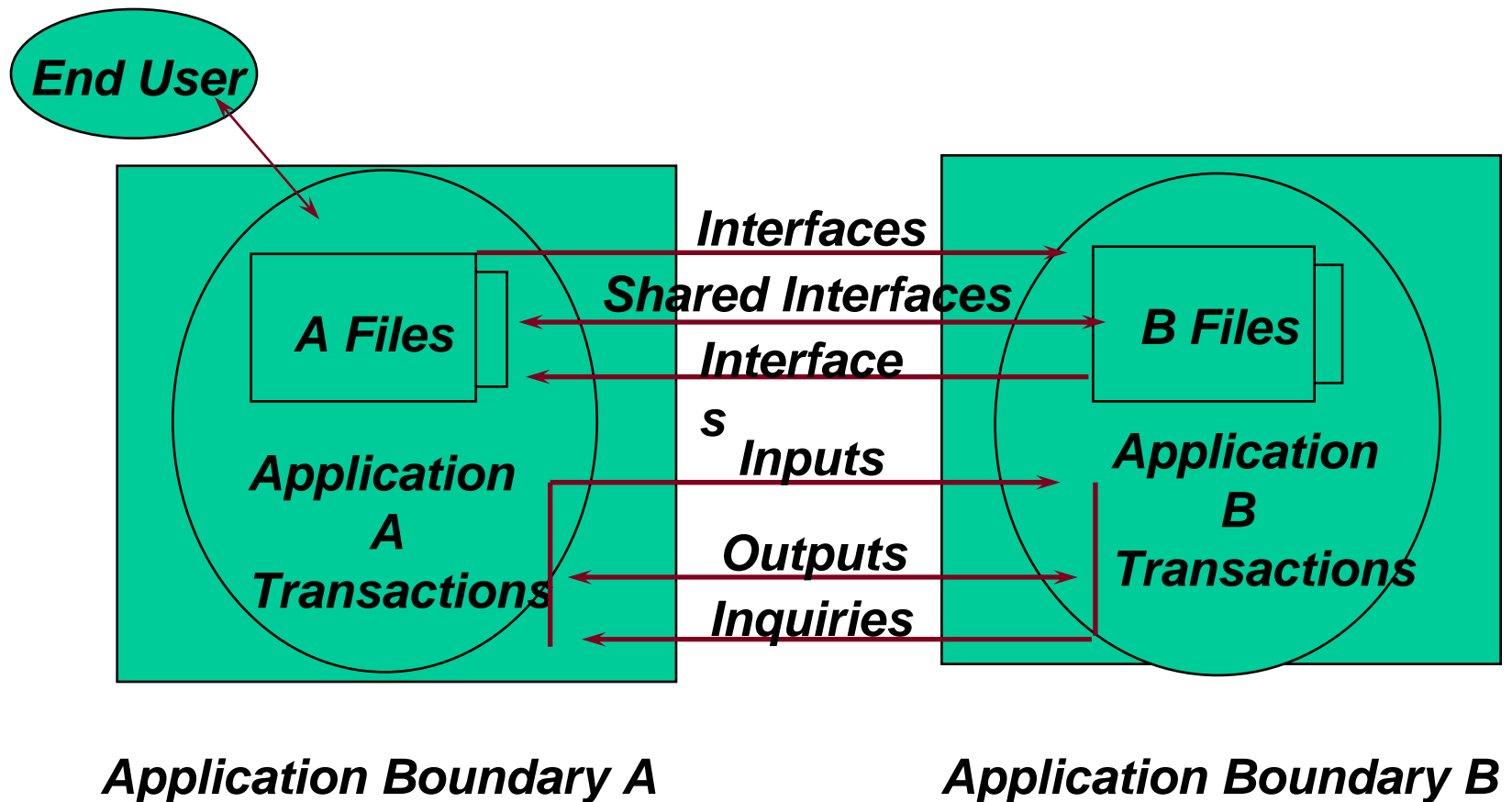


Function Point Software Sizing Steps

1. Establish the subsystem boundaries
2. Identify data functions & determine functional complexity
3. Identify transactional functions & determine functional complexity
4. Calculate the unadjusted Function Point count
5. Determine the value adjustment factor using general system characteristics
6. Calculate the adjusted function point count

Function Point Sizing

Step #1 – Establish Application Boundary



Function Point Sizing

Step #2A – Identify Data Functions

- Internal Logical Files (ILF)
- External Interface Files (EIF)





Internal Logical Files (ILF)

- Logical groupings of data
 - Databases or data sets, master files, tables
- Maintained by an end user
- Utilized within application boundary
- Represent application's maintainable data storage requirements
- Exist when software in use
- Dynamic, not hard coded
- Not considered ILF's: temporary files, work files, sort files; backups, recovery or archive files (unless legal or regulatory required)



External Interface Files (EIF)

- Externally maintained logical groups of data
- Data resides in another system (each EIF is an ILF in another system)
- Outside the application boundary
- User of system being counted requires data for reference purposes only
- Examples: help messages, error messages, reference data
- Not considered EIF's: transaction data, data format/processing



Function Point Sizing

Step #2B – Determine complexity of data functions

- Assign functional complexity based on:
 - Data Element Types (DET)
 - Unique user recognizable, non-repeating fields or attributes of ILF or EIF
 - Record Element Types (RET)
 - Subgroups of data elements within ILF or EIF
- IFPUG complexity matrices
 - Average, Low, or High

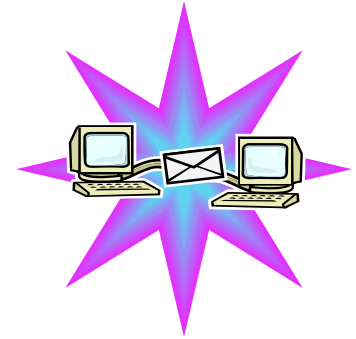
ILF/EIF Complexity Matrix

	<i>1 to 19 DET</i>	<i>20 to 50 DET</i>	<i>51 or more DET</i>
<i>1 RET</i>	<i>L</i>	<i>L</i>	<i>A</i>
<i>2 to 5 RET</i>	<i>L</i>	<i>A</i>	<i>H</i>
<i>6 OR MORE RET</i>	<i>A</i>	<i>H</i>	<i>H</i>
<i>RET = Record Element Type DET = Data Element Type</i>	<i>Functional Complexity: L = Low A = Average H = High</i>		

Function Point Sizing

Step #3A – Identify Transactional Functions

- External Input (EI)
- External Output (EO)
- External Inquiries (EQ)





Transactional Functions

- External Inputs (EI)
 - Maintains internally stored data coming into the application boundary
 - Gives the user the capability to add, change, or delete data contents (e.g., mouse, touch screen, sensor)
- External Outputs (EO)
 - Data passing out of application boundary
 - Gives user ability to produce outputs (e.g., reports, graphics, displays)
- External Inquiries (EQ)
 - Combination of input (request) and output (retrieval)
 - Allows user to select and display specific data from files



Function Point Sizing

Step #3B – Determine complexity of transactional functions

- Assign functional complexity based on:
 - Data Element Types (DET)
 - Unique user recognizable, non-repeating fields or attributes of ILF or EIF
 - File Types Referenced
 - # of ILFs read, referenced, or maintained + # of EIFs read or referenced by the transaction
- IFPUG complexity matrices
 - Average, Low, or High

EI Complexity Matrix

	1 to 4 DET	5 to 15 DET	16 or more DET
0 to 1 FTR	L	L	A
2 FTR	L	A	H
3 or more FTR	A	H	H
FTR = File Type Referenced DET = Data Element Type Functional Complexity: L = Low A = Average H = High			

EO Complexity Matrix

	<i>1 to 5 DET</i>	<i>6 to 19 DET</i>	<i>20 or more DET</i>
<i>0 to 1 FTR</i>	<i>L</i>	<i>L</i>	<i>A</i>
<i>2 to 3 FTR</i>	<i>L</i>	<i>A</i>	<i>H</i>
<i>4 or more FTR</i>	<i>A</i>	<i>H</i>	<i>H</i>
<i>FTR = File Type Referenced</i> <i>DET = Data Element Type</i> <i>Functional Complexity: L = Low</i> <i>A = Average</i> <i>H = High</i>			

EQ Complexity Matrix

	<i>1 to 5 DET</i>	<i>6 to 19 DET</i>	<i>20 or more DET</i>
<i>0 to 1 FTR</i>	<i>L</i>	<i>L</i>	<i>A</i>
<i>2 to 3 FTR</i>	<i>L</i>	<i>A</i>	<i>H</i>
<i>4 or more FTR</i>	<i>A</i>	<i>H</i>	<i>H</i>
<i>FTR = File Type Referenced DET = Data Element Type Functional Complexity: L = Low A = Average H = High</i>			



Function Point Software Sizing

Step #4 – Calculate Unadjusted Function Point Count

- Group Data and Transactional function types by
 - Low
 - Average
 - High
- Calculate Unadjusted Function Point Count
 - $\sum \{ \text{Function Type QTY} \times \text{Functional Complexity} \}$

Function Point Software Sizing

ILF Calculation Example

<i>Function Type</i>	<i>Function Type QTY</i>	<i>Functional Complexity</i>	<i>Complexity Totals</i>	<i>Function Type Totals</i>
ILF	3	Low	$x 7 = 21$	96
	3	Average	$x 10 = 30$	
	3	High	$x 15 = 45$	



Function Point Software Sizing

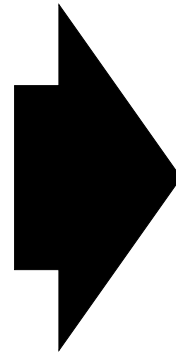
Step #5 – Determine Value Adjustment Factor

- Adjust the Function Point Count for system's technical and operational characteristics
 - Accounts for additional functions due to System Level requirements
- Judge the “degrees of influence” imposed by 14 General Systems Characteristics
 - Rate each characteristic from 0 (no influence) through 5 (extremely strong influence)
 - Sum the ratings to derive Total Degrees of Influence (TDI)

Value Adjustment Factor

General System Characteristics

*Data Communications
Distributed Processing
Performance
Heavily Used Configuration
Transaction Rates
On-Line Data Entry
Design for End User Efficiency
On-Line Update
Complex Processing
Usable In Other Applications
Installation Ease
Operational Ease
Multiple Sites
Facilitate Change*



Degrees of Influence

*0 Not Present
1 Incidental Influence
2 Moderate Influence
3 Average Influence
4 Significant Influence
5 Strong Influence
Throughout*

Value Adjustment Factor

Example – Transaction Rate

<i>Score</i>	<i>Meaning</i>
0	No peak transaction period anticipated
1	Peak transaction period (e.g. monthly, quarterly, annually, seasonally) anticipated
2	Weekly peak transaction period anticipated
3	Daily peak transaction period anticipated
4	Transaction rate(s) stated by the user in the application requirements or service level agreements are high enough to require performance analysis tasks in the design phase
5	Transaction rate(s) stated by the user in the application requirements or service level agreements are high enough to require both performance analysis tasks and use of performance analysis tools in design, development and/or installation phases



Function Point Software Sizing

Step #5 – Calculate Adjusted Function Point Count

$$FPC = UFPC \times ((TDI \times 0.01) + 0.65)$$

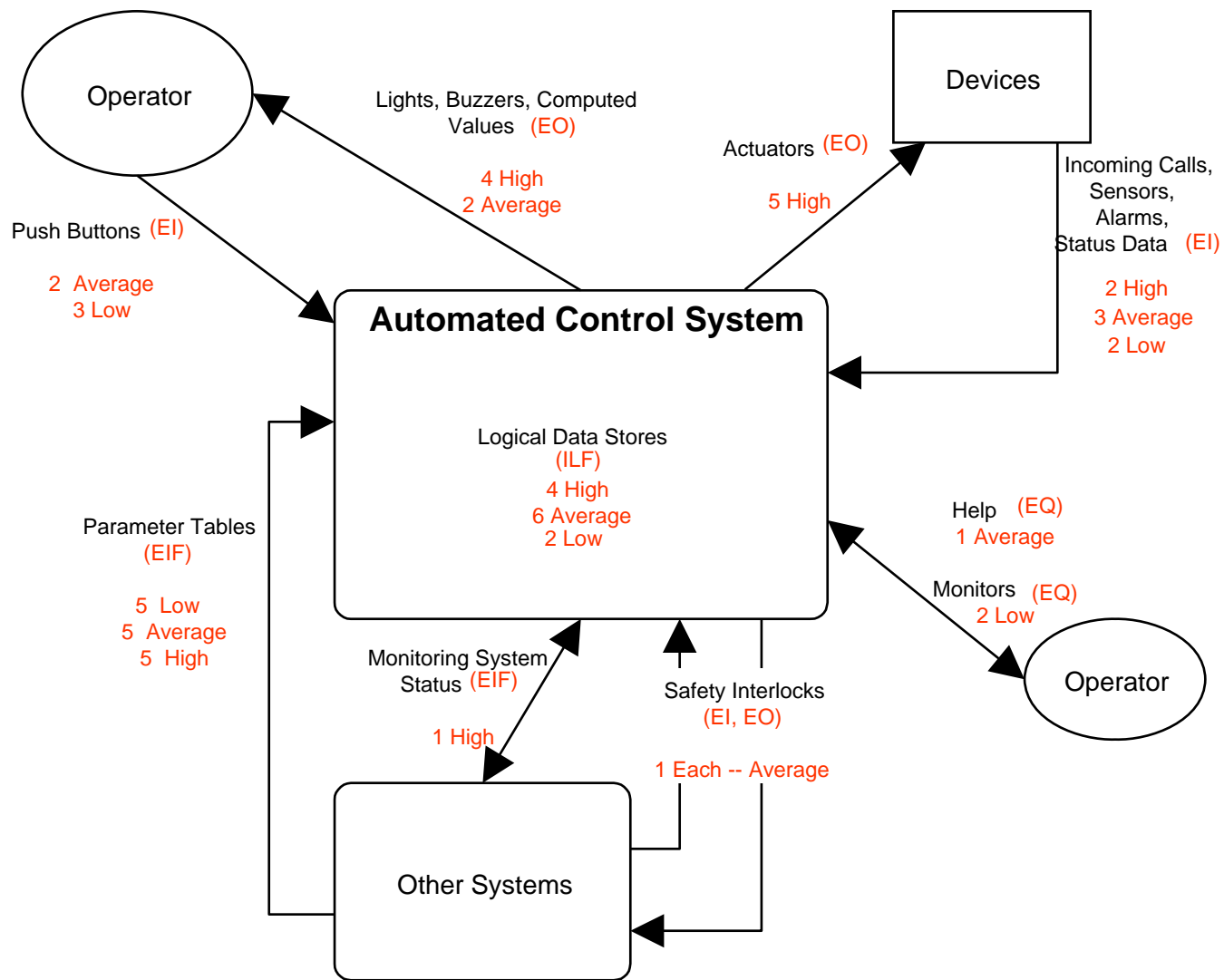
Where UFPC = Unadjusted Function Point Count
= (Function/Type Counts) x (Complexity Weights)

TDI = Total Degree of Influence
= Sum of Degrees of Influence for 14 categories

Function Point Sizing Screen

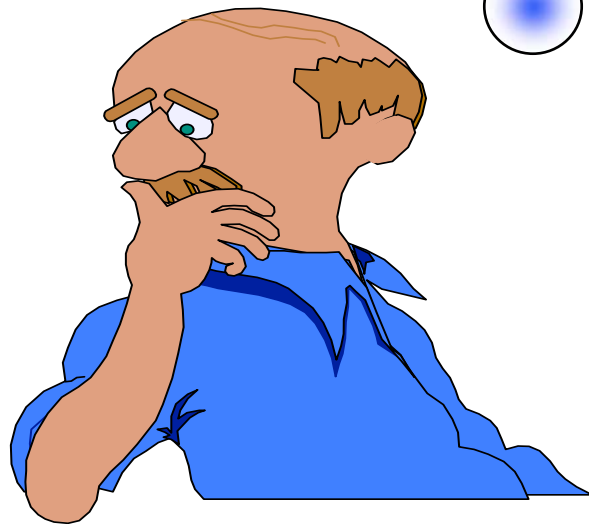
Function Point Sizing					Value Adjustment Factor Calculation		
Unadjusted Function Point Calculation					General System Characteristics		
Function Type	Functional Complexity	Complexity Totals	Function Type Totals		Degree of Influence		
Internal Logical Files	<input type="text" value="7"/>	Low * 7 =	49	64	1. Data Communications	<input type="text" value="2"/>	
	<input type="text" value="0"/>	Avg. * 10 =	0		2. Distributed Processing	<input type="text" value="0"/>	
	<input type="text" value="1"/>	High * 15 =	15		3. Performance	<input type="text" value="1"/>	
External Interface Files	<input type="text" value="0"/>	Low * 5 =	0	21	4. Heavily Used Configuration	<input type="text" value="0"/>	
	<input type="text" value="3"/>	Avg. * 7 =	21		5. Transaction Rates	<input type="text" value="5"/>	
	<input type="text" value="0"/>	High * 10 =	0		6. On-Line Data Entry	<input type="text" value="3"/>	
External Inputs	<input type="text" value="2"/>	Low * 3 =	6	22	7. Design for End User Efficiency	<input type="text" value="0"/>	
	<input type="text" value="4"/>	Avg. * 4 =	16		8. On-line Update	<input type="text" value="2"/>	
	<input type="text" value="0"/>	High * 6 =	0		9. Complex Processing	<input type="text" value="0"/>	
External Outputs	<input type="text" value="1"/>	Low * 4 =	4	9	10. Usable in Other Applications	<input type="text" value="1"/>	
	<input type="text" value="1"/>	Avg. * 5 =	5		11. Installation Ease	<input type="text" value="0"/>	
	<input type="text" value="0"/>	High * 7 =	0		12. Operational Ease	<input type="text" value="4"/>	
External Inquires	<input type="text" value="2"/>	Low * 3 =	6	18	13. Multiple Sites	<input type="text" value="0"/>	
	<input type="text" value="0"/>	Avg. * 4 =	0		14. Facilitate Change	<input type="text" value="2"/>	
	<input type="text" value="2"/>	High * 6 =	12		Total Degree of Influence	20	
Unadjusted Function Point Count			134	Bulkiness	<input type="text" value="1.000"/>	SIZE	114
Adjusted Function Point Count			114	<input type="button" value="OK"/> <input type="button" value="Cancel"/>			

Function Point Example





Who ya going to call?



***IFPUG Executive Office
191 Clarksville Road
Princeton Junction, NJ 08550
Phone: 609-799-4900
Fax: 609-799-7032
ifpug@ifpug.org***

www.ifpug.org