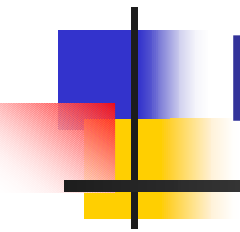
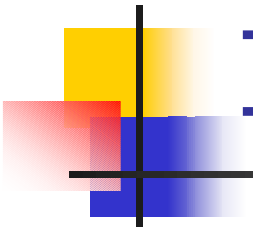


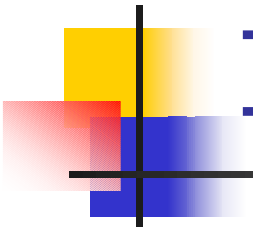
Function Point Analysis

By: Abbas HeydarNoori



Introduction

- Function point metrics, developed by Alan Albercht of IBM, were first published in 1979
- In 1984, the International Function Point Users Group (IFPUG) was set up to clarify the rules, set standards, and promote their use and evolution



Introduction (Cont'd)

- Function point metrics provide a standardized method for measuring the various functions of a software application.
- Function point metrics, measure functionality from the users point of view, that is, on the basis of what the user requests and receives in return

Introduction (Cont'd)

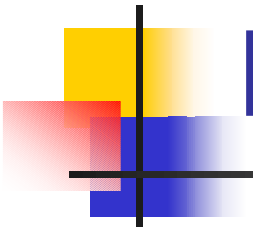
- Albercht's initial definition:
  - This gives a dimensionless number defined in function points which we have found to be an effective relative measure of function value delivered to our customer

Objectives of FPA

- Function point analysis measures software by quantifying the functionality the software provides to the user based primarily on logical design. With this in mind, the objectives of function point analysis are to:
  - Measure functionality that the user requests and receives
  - Measure software development and maintenance independently of technology used for implementation
- In addition to meeting the above objectives, the process of counting function points should be:
  - Simple enough to minimize the overhead of the measurement process
  - A consistent measure among various projects and organizations

Benefits of FPA

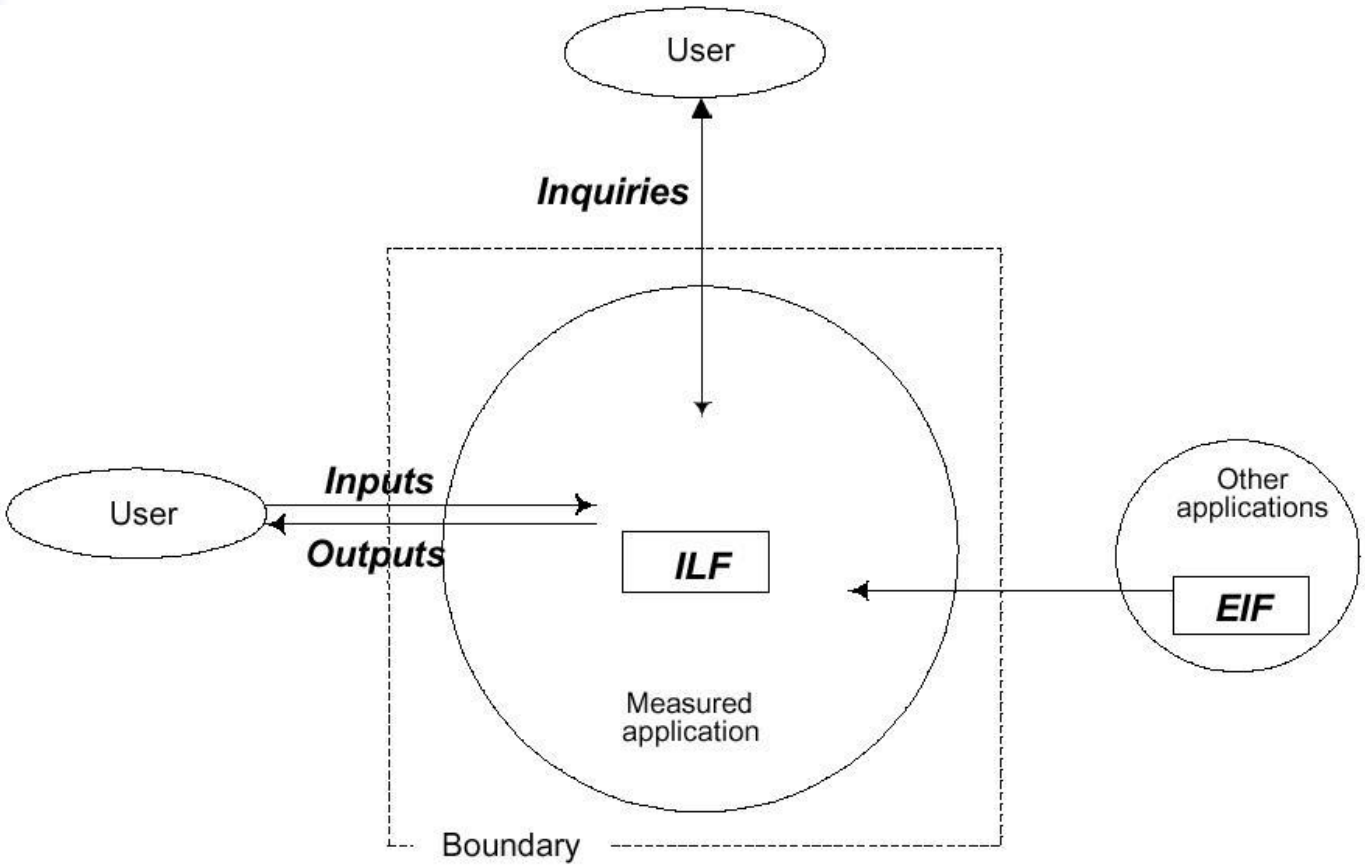
- Organizations can apply function point analysis as:
  - A tool to determine the size of a purchased application package by counting all the functions included in the package
  - A tool to help users determine the benefit of an application package to their organization by counting functions that specifically match their requirements
  - A tool to measure the units of a software product to support quality and productivity analysis
  - A vehicle to estimate cost and resources required for software development and maintenance
  - A normalization factor for software comparison

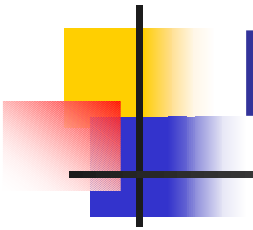


FPA Overview

- The first step in calculating FP is to identify the counting boundary.
  - Counting boundary: The border between the application or project being measured and external applications or the user domain.
  - A boundary establishes which functions are included in the function point count

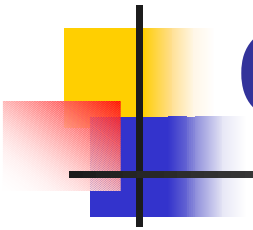
FPA Components





FPA Overview (Cont'd)

- The next step is determining the unadjusted function point (UFP) count
- UFP reflects the specific countable functionality provided to the user by the project or application



Calculation of the UFP

- This calculation begins with the counting of the five function types of a project or application:
  - Two data function types
  - Three transactional function types

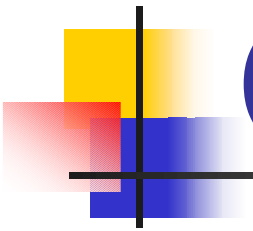
Data Function Types

- ***Internal Logical File (ILF)***: a user identifiable group of logically related data or control information maintained within the boundary of the application
- ***External Interface File (EIF)***: a user identifiable group of logically related data or control information referenced by the application, but maintained within the boundary of another application.
  - This means that EIF counted for an application, must be an ILF in another application

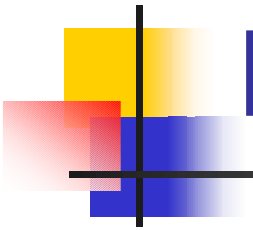
Transactional Function Types

- ***External Input (EI)***: An EI processes data or control information that comes from outside the application's boundary. The EI is an elementary process.
  - Elementary process: The smallest unit of activity that is meaningful to the end user in the business

Transactional Function Types (Con'd)

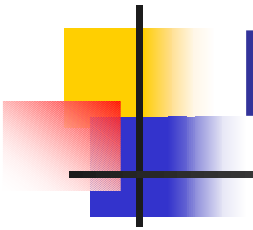


- ***External Output (EO)***: An EO is an elementary process that generates data or control information sent outside the application's boundary
- ***External Inquiry (EQ)***: An EQ is an elementary process made up of an *input-output* combination that results in data retrieval



FPA Overview (Con'd)

- These 5 function types are then ranked according to their complexity: Low, Average or High, using a set of prescriptive standards.
- Organizations that use FP methods, develop criteria for determining whether a particular entry is Low, Average or High.
- Nonetheless, the determination of complexity is somewhat subjective.

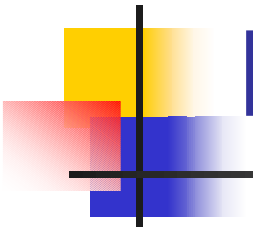


FPA Overview (Con'd)

- After classifying each of the five function types, the UFP is computed using predefined weights for each function type

UFP Calculation Table

Function Type	Functional Complexity		Complexity Totals	Function Type Totals
ILFs	_____	Low	X 7 = _____	
	_____	Average	X 10 = _____	
	_____	High	X 15 = _____	
				_____
EIFs	_____	Low	X 5 = _____	
	_____	Average	X 7 = _____	
	_____	High	X 10 = _____	
				_____
EIs	_____	Low	X 3 = _____	
	_____	Average	X 4 = _____	
	_____	High	X 6 = _____	
				_____
EOs	_____	Low	X 4 = _____	
	_____	Average	X 5 = _____	
	_____	High	X 7 = _____	
				_____
EQs	_____	Low	X 3 = _____	
	_____	Average	X 4 = _____	
	_____	High	X 6 = _____	
				_____
		Total Unadjusted Function Point Count		_____



FPA Overview (Con'd)

- The last step involves assessing the environment and processing complexity of the project or application as a whole.
- In this step, the impact of 14 general system characteristics is rated on a scale from 0 to 5 in terms of their likely effect on the project or application

Value Adjustment Factor (VAF) Calculation Table

- 0 = No Influence
- 1 = Incidental
- 2 = Moderate
- 3 = Average
- 4 = Significant
- 5 = Essential

General System Characteristics (GSCs)	Degree of Influence (DI) 0 - 5
1. Data Communications	_____
2. Distributed Data Processing	_____
3. Performance	_____
4. Heavily Used Configuration	_____
5. Transaction Rate	_____
6. Online Data Entry	_____
7. End-User Efficiency	_____
8. Online Update	_____
9. Complex Processing	_____
10. Reusability	_____
11. Installation Ease	_____
12. Operational Ease	_____
13. Multiple Sites	_____
14. Facilitate Change	_____
Total Degree of Influence (TDI)	_____
Value Adjustment Factor (VAF)	_____
	VAF = (TDI * 0.01) + 0.65

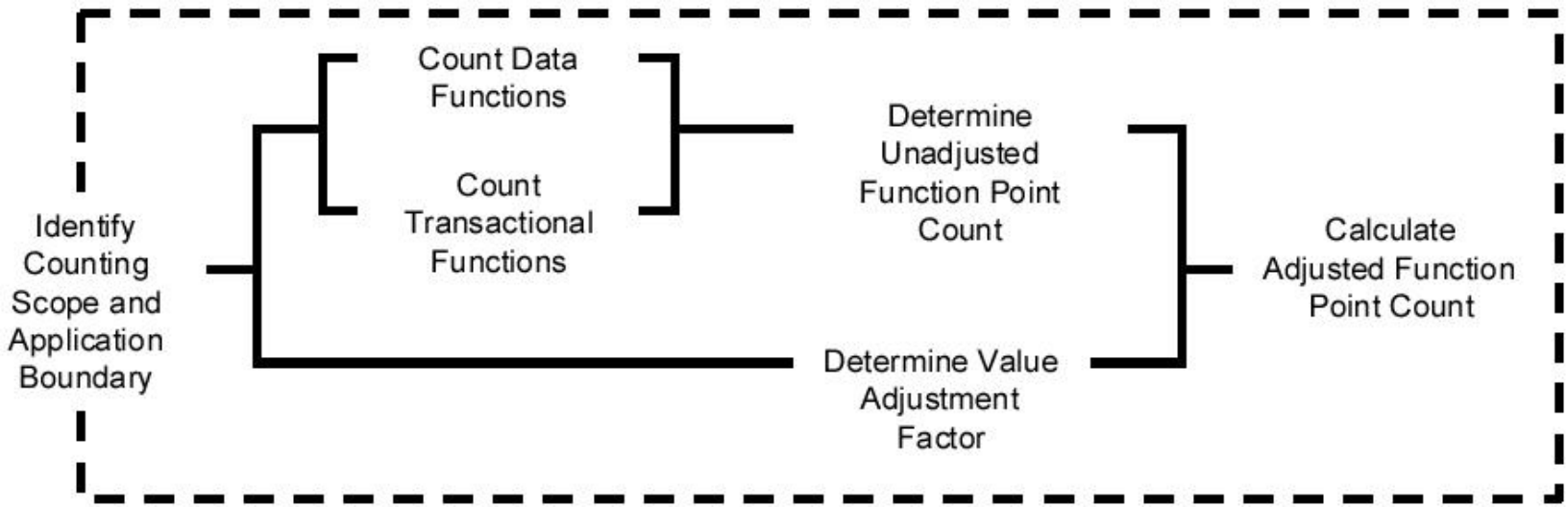
FPA Overview (Con'd)

- On the whole:

$$FP = UFP \times VAF$$

- The constant values in the equation and the weighting factors are determined empirically

FPA Procedure at a Glance

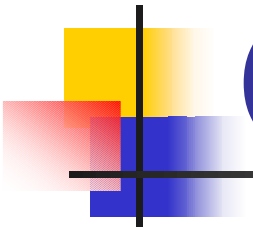


Evolution of Function Points

- Over the years, various improvements have been made to the 1979 initial description and successive versions have been published

Version	Reference
Albrecht 79	Albrecht, 1979
Albrecht 83	Albrecht and Gaffney, 1983
GUIDE 85	GUIDE, 1985
IFPUG 86	International Function Point Users Group, 1986
IFPUG 88	International Function Point Users Group, 1988
IFPUG 90	International Function Point Users Group, 1990

Evolution of Function Points (Cont'd)



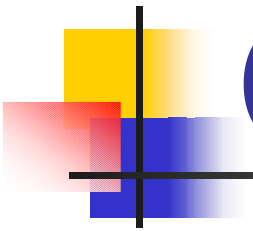
- The first 3 versions addressed the structure of FP
- The 3 IFPUG versions addressed the clarification of the rules and guidelines
- The Albercht 79 model, had 4 function types and one set of weights and 10 general system characteristics for VAF

Albercht 79 model

Albrecht 79	
Function Types	Weights
Files	10
Inputs	4
outputs	5
Inquiries	4

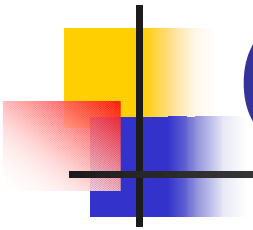
General System Characteristics **(GSC)**
Albrecht 79
1. Backup 2. Data communications 3. Distributed processing 4. Performance issues 5. Heavily used configuration 6. Online data entry 7. Conversational data entry 8. Online update of master files 9. Complex functions 10. Internal processing complex
Value adjustment factor = (0.75 $\pm$ 25% max)

Evolution of Function Points (Cont'd)

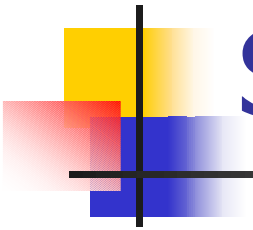


- The Albercht 83 model, was expanded to 5 function types, 3 sets of weights and 14 system characteristics

Evolution of Function Points (Cont'd)



- The GUIDE 85, introduced a new dimension to function points through a set of rules for the functional complexity rating (Low, Average and High) of the five function types.
- The function types were decomposed into 3 types of primary components and 2 dimensional matrices with pre-determined ranges of values were used for rating purposes
  - This allows consistent rating across individuals and organizations.



Some Terms

- ***DET***: a *data element type* is a unique user recognizable, non-repeated field. For example, an account number that is stored in multiple fields is counted as one DET.
- ***RET***: A *record element type* is a user recognizable subgroup of data elements within an ILF or EIF.

Some Terms (Cont'd)

- ***FTR:*** A *file type referenced* is
  - An internal logical file read or maintained by a transactional function or
  - An external interface file read by a transactional function

Matrix Used for ILF and ELF

	1 to 19 DET	20 to 50 DET	51 or more DET
1 RET	Low	Low	Average
2 to 5 RET	Low	Average	High
6 or more RET	Average	High	High

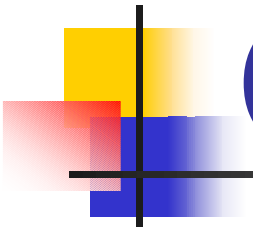
Matrix Used for EI

	1 to 4 DET	5 to 15 DET	16 or more DET
0 to 1 FTR	Low	Low	Average
2 FTRs	Low	Average	High
3 or more FTRs	Average	High	High

Matrix Used for EO and EQ

	1 to 5 DET	6 to 19 DET	20 or more DET
0 to 1 FTR	Low	Low	Average
2 to 3 FTRs	Low	Average	High
4 or more FTRs	Average	High	High

Evolution of Function Points (Cont'd)



- The subsequent versions published by IFPUG have provided further clarification of the rules, guidelines and criteria.
- But, they have not introduced any change to the structure of function point methodology itself

An Example

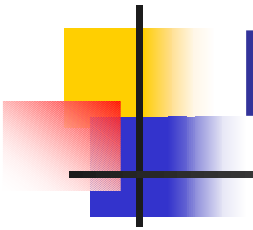
Function Type	Estimated Count	Weight	FP-Count
EI	24	(Average) 4	96
EO	16	(Average) 5	80
EQ	22	(Average) 4	88
ILF	4	(Average) 10	40
ELF	2	(Average) 7	14
UFP count			**318**

An Example (Cont'd)

VAF = 52 * 0.01 + 0.65
= 1.17

$FP_{estimated}$ = 318 x 1.17
= 372

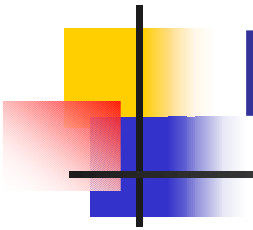
General System Characteristics (GSCs)	Degree of Influence (DI) 0 - 5
1. Data Communications	2
2. Distributed Data Processing	0
3. Performance	5
4. Heavily Used Configuration	5
5. Transaction Rate	2
6. Online Data Entry	4
7. End-User Efficiency	3
8. Online Update	5
9. Complex Processing	4
10. Reusability	5
11. Installation Ease	4
12. Operational Ease	3
13. Multiple Sites	4
14. Facilitate Change	5
Total Degree of Influence (TDI)	52
Value Adjustment Factor (VAF)	1.17



Problems of FPA

- FPA has been criticized as not being universally applicable to all types of software.
  - For example, FPA doesn’t capture all functional characteristics of real-time software

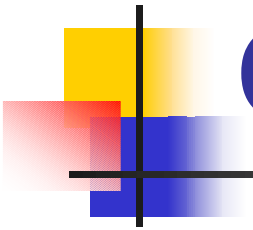
Problems of FPA (Con'd)



- FP metrics are derived from a set of steps, rules and formulas. So they are algorithmic metrics and so, have these problems:
  - Algorithmic metrics are difficult to interpret and the reasons for the assignments of specific weights are not clear

Problems of FPA (Con'd)

- The value of the output of the formula is useful only if the formula is based on a solid theory such as physics, but this is not the case for FP
- The FP definition itself, has not been clarified and has generated some confusion among both practitioners and academics
- What is a metric if it is only a number?



Other Variants of FPA

- FP was originally designed to be applied to business information systems applications.
  - So, the data dimension was emphasized.
    - So, FPA was inadequate for many engineering and embedded systems.

Other Variants of FPA (Cont'd)

- ***Feature Point***
  - Is a superset of FP
  - Suitable for real-time, process-control and embedded software applications tend to have high algorithmic complexity
  - This method counts a new software characteristics: "*algorithms*"

Other Variants of FPA (Cont'd)

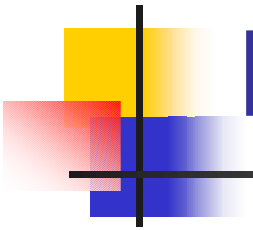
- ***3D Function Point***
  - Developed by Boeing
  - Suitable for applications that emphasize function and control capabilities:
    - *Data dimension*: very similar to basic FP
    - *Functional dimension*: is measured by considering the number of internal operations required to transform input to output data
    - *Control dimension*: is measured by counting the number of transitions between states.
  - Characteristics of all 3 dimensions are counted, quantified and transformed into a measure that provides an indication of the functionality

Other Variants of FPA (Cont'd)

- ***MK II FPA***
  - Developed in the late 80's by Charles Symons in the UK
- ***NESMA***
  - Is a simpler-to-use variant of the IFPUG method
  - Maintained by the Netherlands software metrics association

Other Variants of FPA (Cont'd)

- ***COSMIC-FPP***
  - Is approved by ISO
  - Designed to measure the functional size of real-time, multi-layered software such as used in telecoms, process control and operating systems as well as business applications, all on the same measurement scale
  - Having been developed in the last few years, the method is compatible with modern specification methods such as UML and OO techniques.



References

- A. Abran and P. N. Robillard, ***Function points: a study of their measurement processes and scale transformations***, Journal of Systems and Software, Vol. 25, No. 2, 1994, pp. 171-184
- ***Full Function Points: Counting Practices Manual***, Edited by Software Engineering Laboratory Management Research Laboratory and …, Sep. 1997
- T. Fetcke, ***A Generalized Structure for Function Point Analysis***, In International Workshop on Software Measurement, Lac Supérieur, Québec, Canada, Sep. 1999

References (Con'd)

- http://www.lrgl.uqam.ca/
- ***IFPUG: Function Point Counting Practices Manual***, Release 4.1.1
- R. S. Pressman, ***Software Engineering: A Practitioner's Approach,*** McGraw-Hill, 2000