Software Project Management

Durga Prasad Mohapatra

Professor

CSE Deptt.

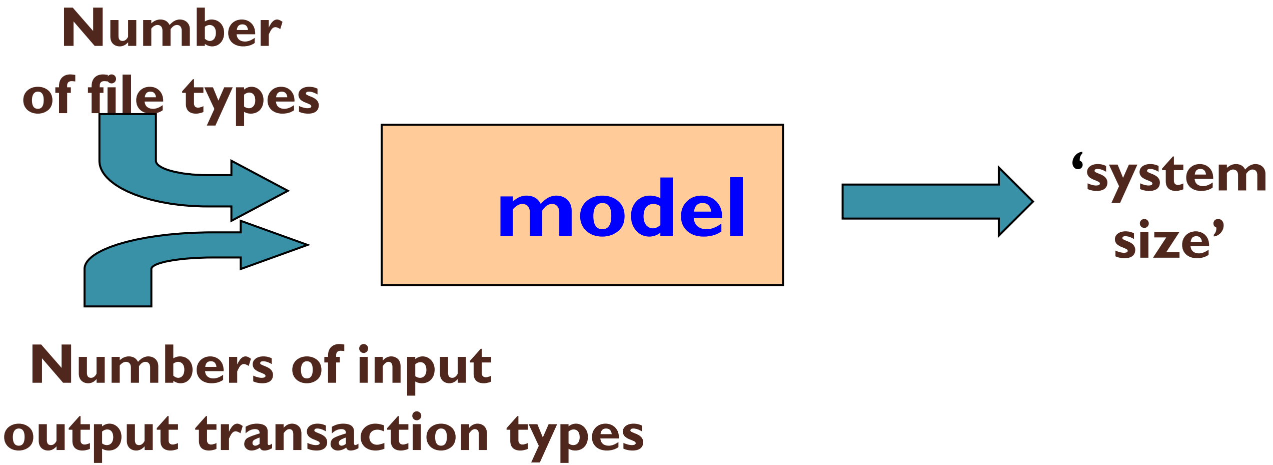
NIT Rourkela

Project Estimation Techniques cont…

- **Parametric models**
  - **Albrecht/IFPUG function points**

Parametric model for Size

- Important example: Function Points
- FPs originally used to estimate Lines of Code, rather than effort



Parametric models for Size

We shall examine four parametric models more closely :

1. Albrecht/IFPUG function points
2. Symons/Mark II function points
3. COSMIC function points
4. COCOMO81 and COCOMO II

International Function Point Users Group (IFPUG)

Purpose

- To promote and encourage use of Function Points
- To develop consistent and accurate counting guidelines

Benefits

- Networking with other counters
- IFPUG Counting Practices Manual
- Research projects
- Hotline
- Newsletter
- Certification

Extent of Usage:

- Member companies include all industry sectors
- Over 1200 members in more than 30 countries

Albrecht/IFPUG function point analysis

- Albrecht worked at IBM:
  - Needed a way of measuring the relative productivity of different programming languages.
  - **Needed some way of measuring the size of an application without counting lines of code.**
- **Identified five parameters:**
  - Counted occurrences of each type of functionality in order to get an indication of the size of an information system
  - It is a top-down approach

Why IFPUG Thinks One Should Not Use LOC?

- Lines of code tend to reward profligate design and penalize concise design
- There is no industry standards (ISO or otherwise) for lines of code
- Lines of code cannot be used for normalizing across platform, language or by organization
- Some 4GL do not even use lines of code
- Lines of code can be misleading.

How Do Function Points Overcome LOC Problems?

- Function points are independent of the language, tools, or methodologies used for implementation
- Function points can be estimated early in analysis and design
- Since function points are based on the user's external view of the system:
  - **Non-technical users of the software have a better understanding of what function points are measuring**

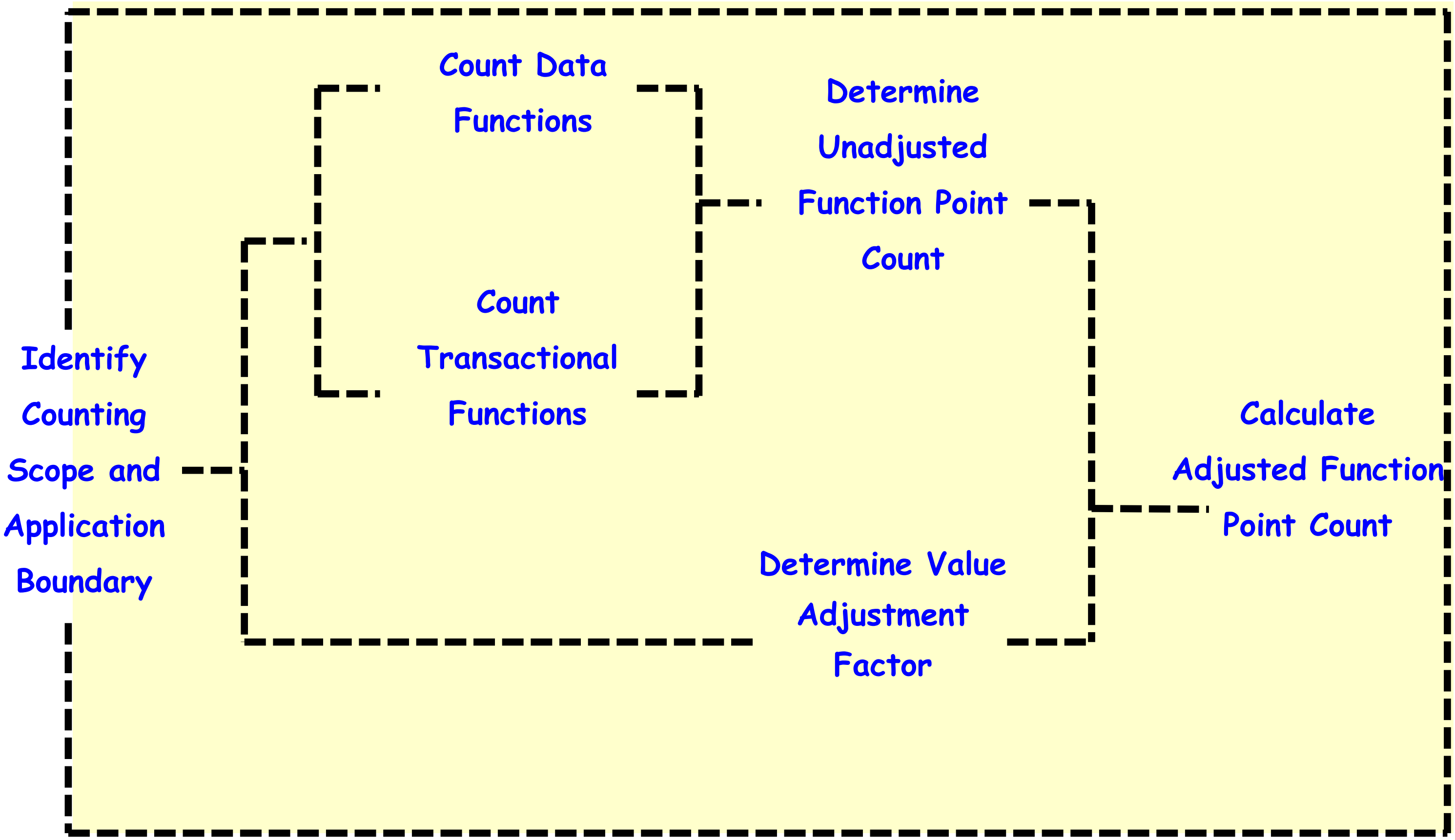
Objectives of Function Point Counting

- Measure functionality that the user requests and receives…
- Measure software development and maintenance independently of technology used for implementation…
- Simple enough to minimize the overhead of the measurement process…
- A consistent measure among various projects and organizations …

When Should You Count Function Points?

- **Early and often**
  - The sooner you can quantify what a project is delivering, the sooner it is under control
- Under IFPUG 4.1, there are rules and guidelines:
  - Make it possible to count function points once the requirements have been finalized
  - The estimate of size in function points can be refined continuously through out the development cycle.
- Function points should be recounted throughout the development process: Can measure scope creep and breakage

FP Computation Steps


Identify Counting Scope and Application Boundary
Count Data Functions
Count Transactional Functions
Determine Unadjusted Function Point Count
Determine Value Adjustment Factor
Calculate Adjusted Function Point Count

Key Components in Function Point Analysis

Five key components (or External User Types) are identified

- External Inputs
- External Outputs
- External Inquiries
- Logical Internal Files
- External Interface Files

Key Components in Function Point Analysis cont…

1. **External input (EI) types:**
   - Transactions which update internal computer files
2. **External output (EO) types:**
   - Transactions which extract and display data from internal computer files.
   - Generally involves creating reports.
3. **External inquiry (EQ) types:**
   - User initiated transactions which provide information but do not update computer files.
   - Normally the user inputs some data that guides the system to the information the user needs.

Key Components in Function Point Analysis cont…

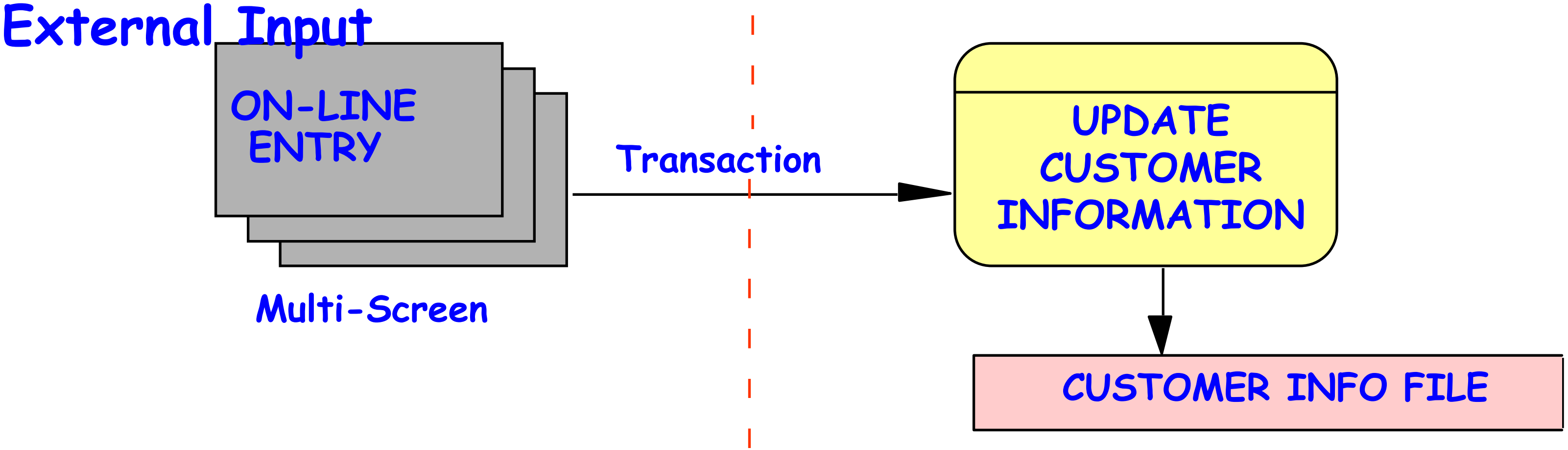
4. **Logical interface file (LIF) types**
   - ○ Equates roughly to a data store in systems analysis terms. Created and accessed by the target system
5. **External interface file types (EIF)**
   - ○ Represents data retrieved from a data store which is actually maintained by a different application.

Function Point Parameters

External Input (Inputs)	Count each unique user data or user control input type that (i) enters the external boundary of the software system being measured and (ii) adds or changes data in a logical internal file.
External Output (Outputs)	Count each unique user data or control output type that leaves the external boundary of the software system being measured.
Internal Logical File (Files)	Count each major logical group of user data or control information in the software system as a logical internal file type. Include each logical file (e.g., each logical group of data) that is generated, used, or maintained by the software system.
External Interface Files (Interfaces)	Files passed or shared between software systems should be counted as external interface file types within each system.
External Inquiry (Queries)	Count each unique input-output combination, where an input causes and generates an immediate output, as an external inquiry type.

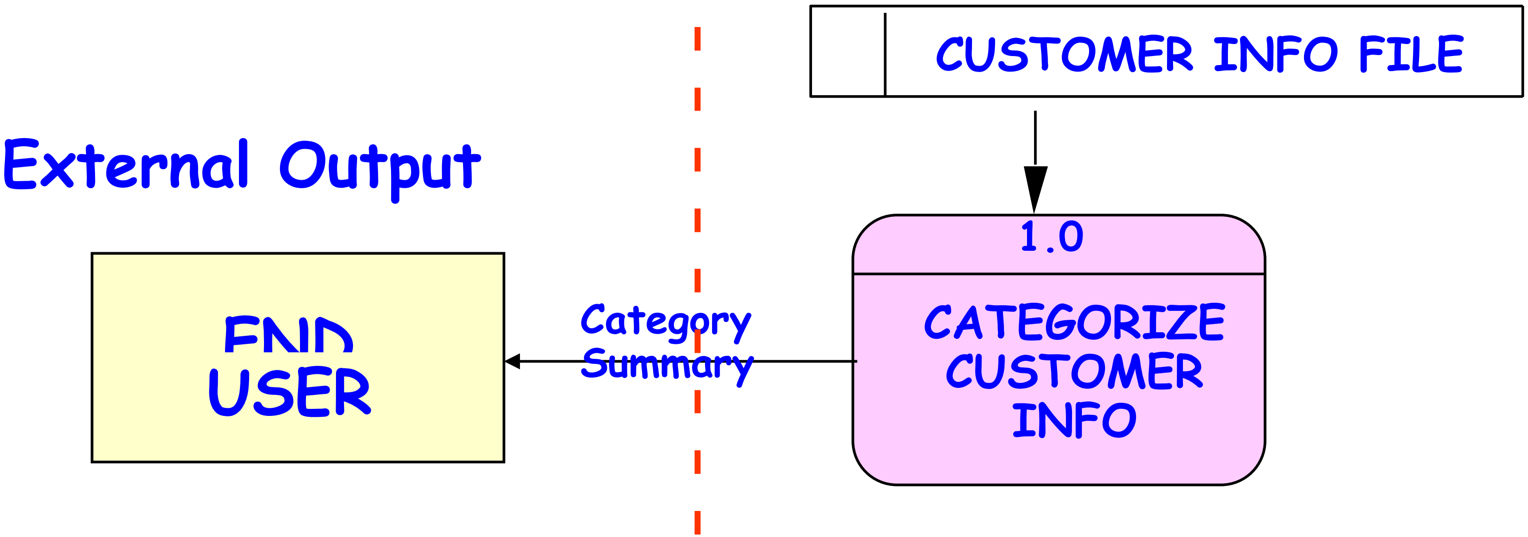
Definition of External input

An External Input (EI) processes data that comes from outside the application's boundary.



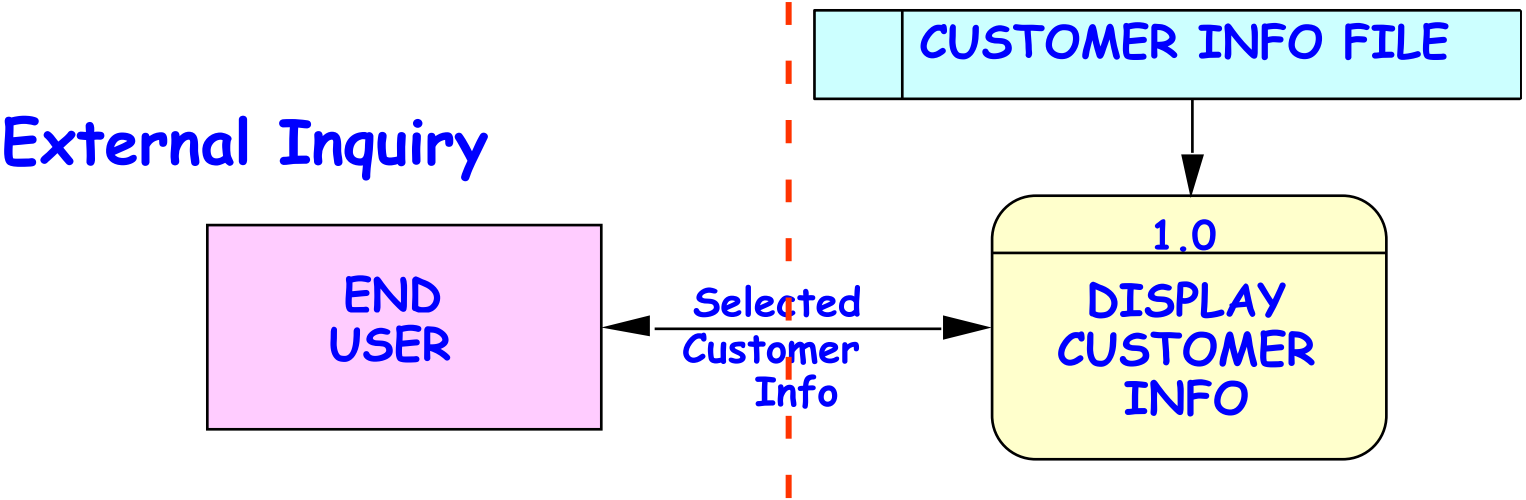
Definition of External Output

An External Output (EO) generates data that is sent outside the application boundary.



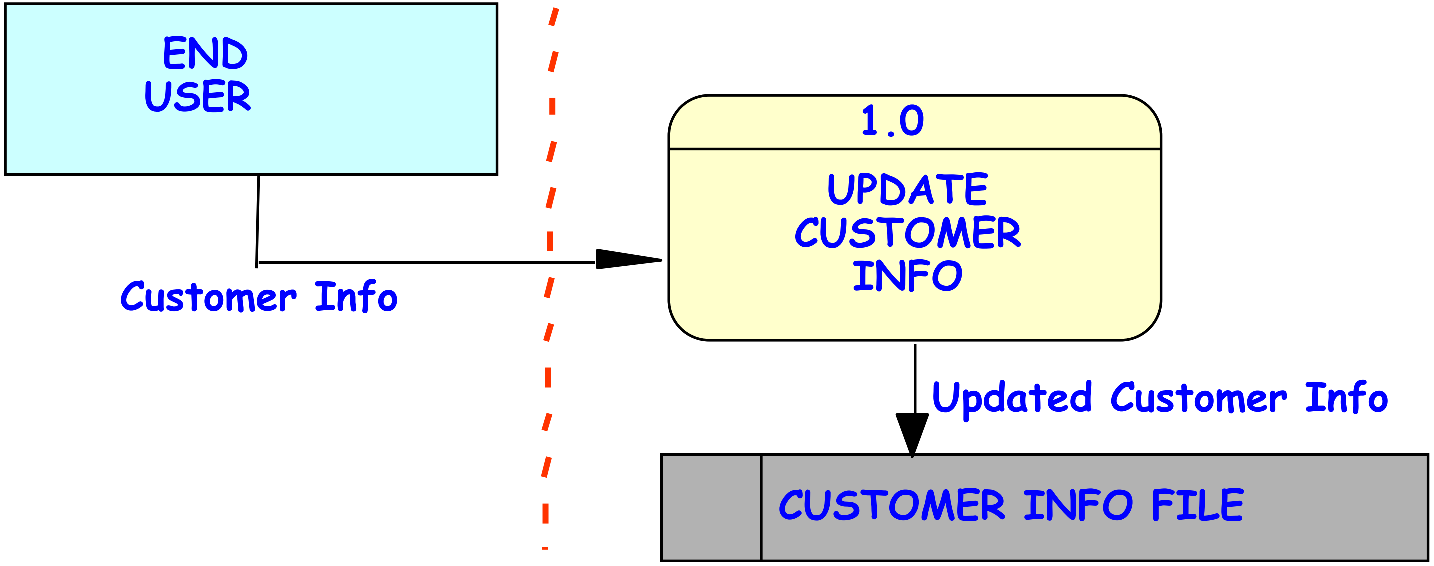
Definition of An Inquiry

An External Inquiry (EQ) is an output that results in data retrieval. The result contains no derived data.



Definition of An IL File

An Internal Logical File (ILF) is a user-identifiable group of logically related data that is maintained within the boundary of the application.

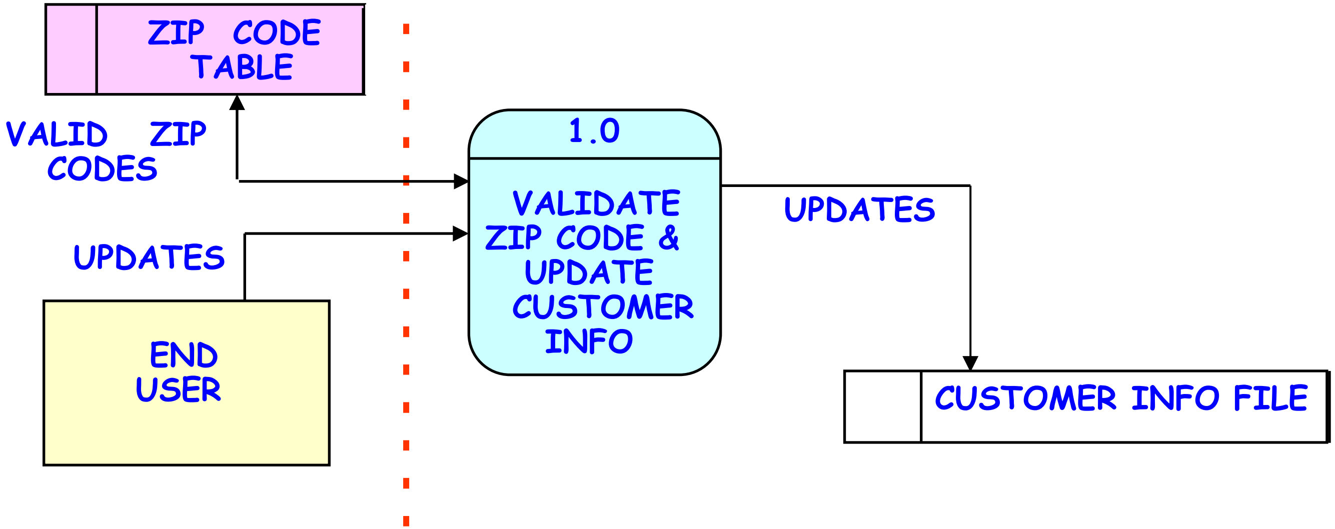


Internal Logical File

Definition of an External Interface File

An External Interface File (EIF) is a user-identifiable group of data referenced by the application, but maintained within the boundary of another application.

External Interface File



Example

- **Place Purchase Order:**
  - **Input data items:**

    Date, Supplier Number,

    Product Code

    Quantity Required

    Date Required
  - **Output data items:**

    PO Number (system generated)
  - **Entities referenced:**

    Product, Purchase Order

    Supplier, Purchase Order Item

Calculating the System Size

- For each function count:
  - Number of input data items n_i
  - Number of output data items n_o
  - Number of entities read/updated n_e
- Add these up for the whole system, giving:
  - Number of input data items N_i
  - Number of output data items N_o
  - Number of entities read/updated N_e

FP Counting - Example

Requirement	inputs	outputs	entity accesses
A1	10	2	4
A2	10	3	6
A3	1	25	1
A4	10	10	9
A5	4	10	5
A6	26	9	2
A7	5	11	8
A8	14	4	5
A9	22	7	4
A10	6	6	4
A11	9	9	7
A12	3	24	5
	Ni = 120	No = 120	Ne = 60

Albrecht complexity multipliers

External user types	Low complexity	Medium complexity	High complexity
External input type	3	4	6
External output type	4	5	7
External inquiry type	3	4	6
Logical internal file type	7	10	15
External interface file type	5	7	10

IFPUG file type complexity

Number of record types	Number of data types		
	< 20	20-50	> 50
1	Low	Low	Average
2 to 5	Low	Average	High
> 5	Average	High	High

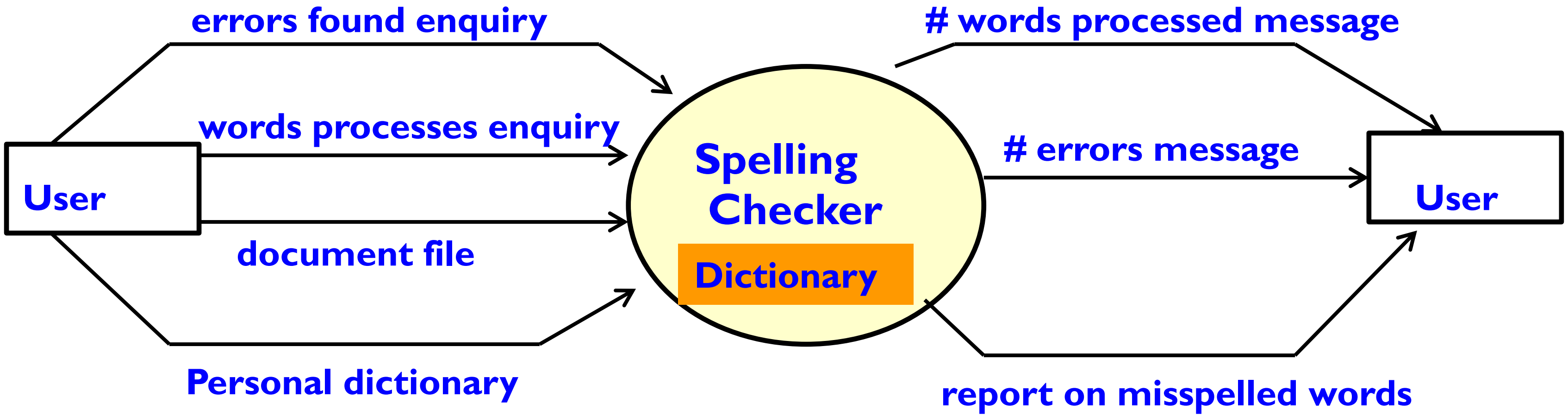
Example

A logical internal file might contain data about purchase orders. These purchase orders might be organized into two separate record types: the main PURCHASE –ORDER details, namely purchase order number, supplier reference and purchase order date, and then details for each PURCHASE-ORDER-ITEM specified in the order, namely the product code, the unit price and number ordered. The number of record types for that file would therefore be 2 and the number of data types would be 6. According to Table 5.3, this file type would be rated as ‘low’. This would mean that according to Table 5.2, the FP count would be 7 for this file.

Example - 1

The Spell-Checker accepts as input a document file and an optional personal dictionary file. The checker lists all words not contained in either of these files. The user can query the number of words processed and the number of spelling errors found at any stage during processing.

Example 1 - cont…



Example 1 - cont…

- 2 users inputs: document file name, personal dictionary name (average)
- 3 users outputs: error report, word count, misspelled error count (average)
- 2 users requests: # processed words?, #spelling errors? (average)
- 1 internal file: dictionary (average)
- 2 external files: document file, personal dictionary (average).

We know that, for average (medium) complexity parameters,

UFP= # inputs*4+ # outputs*5+ # inquiries*4+ # files*10+ # interfaces*7

= 2×4 + 3 × 5 + 2 × 4 + 1 × 10 + 2 × 7

= 55

Function Point: Refinement

14 General Systems Characteristics are evaluated and used to compute a Value Adjustment Factor (VAF)

General System Characteristics

Data Communication
Distributed Data Processing
Performance Objectives
Heavily Used Configuration
Transaction Rate
On-Line Data Entry
End-User Efficiency

On-Line Update
Complex Processing
Reusability
Conversion & Install Ease
Operational Ease
Multiple-Site Use
Facilitate Change

The final calculation is based upon the Unadjusted FP count X VAF

Degrees of Influence

0	Not present, or no influence
1	Incidental influence
2	Moderate influence
3	Average influence
4	Significant influence
5	Strong influence throughout

Procedures to Determine the VAF

- Evaluate each of the 14 general system characteristics on a scale from zero to five to determine the degree of influence (DI)
- Add the degrees of influence for all 14 general system characteristics to produce the total degree of influence (TDI).
- Insert TDI into the following equation to produce the Value Adjustment Factor (VAF) / Technical Complexity Factor (TCF).

$$\mathbf{VAF = (TDI * 0.01) + 0.65}$$

- It expresses the overall impact of the corresponding parameter on the development effort.

Procedures to Determine the VAF

The following VAF is calculated, if the degree of influence (DI) for each of the 14 GSC descriptions is 3, (i.e. 3*14):

VAF = (42 * 0.01) + 0.65 = 1.07

The Adjusted Function Point is calculated as follows:

FP=UFP* VAF

= 55*1.07

= 58.85

Procedures to Determine the VAF cont ..

14 general system characteristics with different DIs:

1.	Data Communication	3
2.	Distributed Data Processing	0
3.	Performance Criteria	4
4.	Heavily Utilized Hardware	0
5.	High Transaction Rates	3
6.	Online Data Entry	3
7.	Online Updating	3

Total Degree of Influence (TDI)=36

8.	End-user Efficiency	3
9.	Complex Computations	0
10.	Reusability	3
11.	Ease of Installation	3
12.	Ease of Operation	5
13.	Portability	3
14.	Maintainability	3

Example 1 cont…

So, VAF = (36 * 0.01) + 0.65 = 1.01

The Adjusted Function Point is calculated as follows:

FP = UFP * VAF

= 55 * 1.01

= 55.55

Example 3

A Payroll application has:

1. Transaction to input, amend and delete employee details – an EI that is rated of medium complexity
2. A transaction that calculates pay details from timesheet data that is input – an EI of high complexity
3. A transaction of medium complexity that prints out pay-to-date details for each employee – an EO of medium complexity
4. A file of payroll details for each employee – assessed as of medium complexity LIF
5. A personnel file maintained by another system is accessed for name and address details – a simple EIF

What would be the FP counts for these?

FP counts

1. Medium EI = 4 FPs
2. High complexity EI = 6 FPs
3. Medium complexity EO = 5 FPs
4. Medium complexity LIF = 10 FPs
5. Simple EIF = 5 FPs

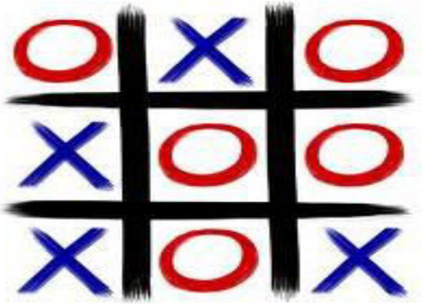
Total (UFP) = 30 FPs

External user types	Low complexity	Medium complexity	High complexity
External input type	3	4	6
External output type	4	5	7
External inquiry type	3	4	6
Logical internal file type	7	10	15
External interface file type	5	7	10

If previous projects delivered 5 FPs a day, implementing the above should take 30/5 = 6 days

Exercise 1:Tic-Tac-Toe Computer Game

- As soon as either of the human player or the computer wins,
  - A message announcing the winner should be displayed.
- If neither player manages to get three consecutive marks along a straight line,
  - And all the squares on the board are filled up,
  - Then the game is drawn.
- The computer always tries to win a game.



O	X	O
X	O	O
X	O	X

Exercise 2

- It is needed to develop an Alumni Repository software for IIM, Ranchi. The software will extract the details of students from the existing academic software of IIM, Ranchi. It will provide an online display of the Alumni names. The details of the Alumni can be entered by any one by double clicking on the Alumni name. The details of Alumni would be stored in a file. It should be possible to print out a report detailing all alumni.
- Determine UFP.

Exercise 3: Supermarket Prize Scheme

- A supermarket needs to develop the following software to encourage regular customers.
- TO register, a customer needs to supply his/her residence address, telephone number, and the driving license number.
- Each customer who registers for this scheme is assigned a unique customer number (CN) by the computer.
- Based on the generated CN, a clerk manually prepares a customer identity card after getting the market manager's signature on it.
- A customer can present his customer identity card to the check out staff when he makes any purchase. In this case, the value of his purchase is credited against his CN.
- At the end of each year, the supermarket intends to award surprise gifts to 10 customers who make the highest total purchase over the year. Also, it intends to award a 22 caret gold coin to every customer whose purchase exceeded Rs.10,000/-.
- The entries against the CN are reset on the last day of every year after the prize winners' lists are generated.

Exercise 4

Calculate Unadjusted Function Point (UFP), Complexity Adjustment Factor (CAF) and Function Point (FP) for the following problem.

Number of user inputs=32

Number of user outputs=60

Number of user inquiries=24

Number of files=8

Number of external interfaces=2

Assume all weighting factors to be average and all complexity adjustment values to be average.

Exercise 5

Calculate Unadjusted Function Point (UFP), Complexity Adjustment Factor (CAF) and Function Point (FP) for the following problem.

Number of user inputs=24 (Weighting factor is average)

Number of user outputs=46 (Weighting factor is simple)

Number of user inquiries=8 (Weighting factor is complex)

Number of files=4 (Weighting factor is average)

Number of external interfaces=2 (Weighting factor is simple)

The various complexity adjustment values are 4, 1, 0, 3, 3, 5, 4, 4, 3, 3, 2, 2, 4, 5.

Summary

- Discussed Albrecht/IFPUG function points analysis
- Also, discussed the steps to compute FP count
- Presented Albrecht/IFPUG function point parameters
- Explained Albrecht/IFPUG function points counting with some suitable examples.

References :

1. B. Hughes, M. Cotterell, R. Mall, *Software Project Management*, Sixth Edition, McGraw Hill Education (India) Pvt. Ltd., 2018.
2. R. Mall, *Fundamentals of Software Engineering*, Fifth Edition, PHI Learning Pvt. Ltd., 2018.

Thank you