CTIS359
Principles of Software Engineering

Introduction to Project Planning & Functional Size Measurement

"One of my most productive days was throwing away 1,000 lines of code."

Ken Thompson

Today

- Project Management
  - Planning
    - Estimation
      - Functional Related Metrics
        - IFPUG's Counting Principles
          - Examples

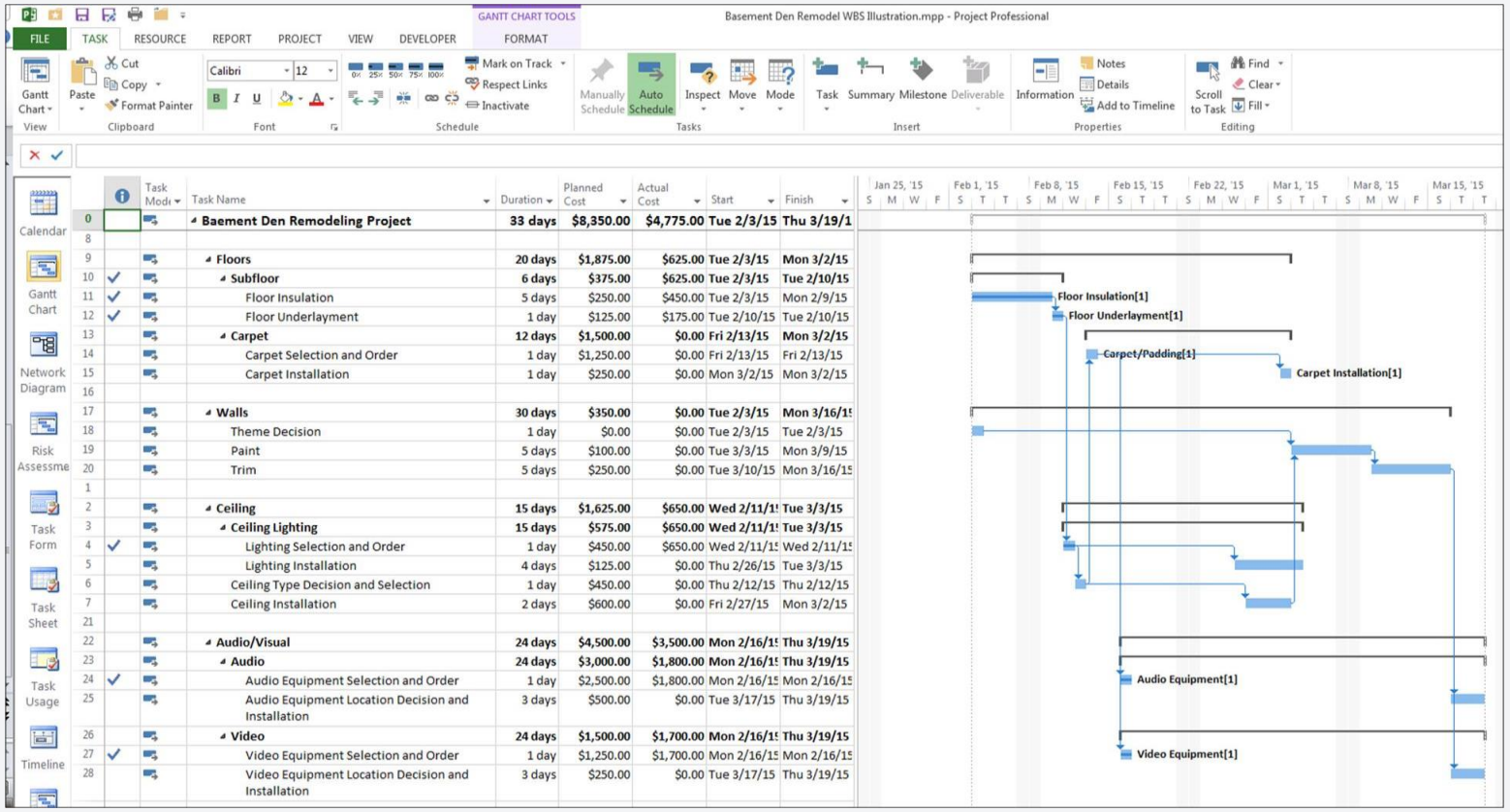
Project Management

- Managing people is NOT the easiest job in the World ☹
- Managing SWE is not the easier than managing non-SWE ☹

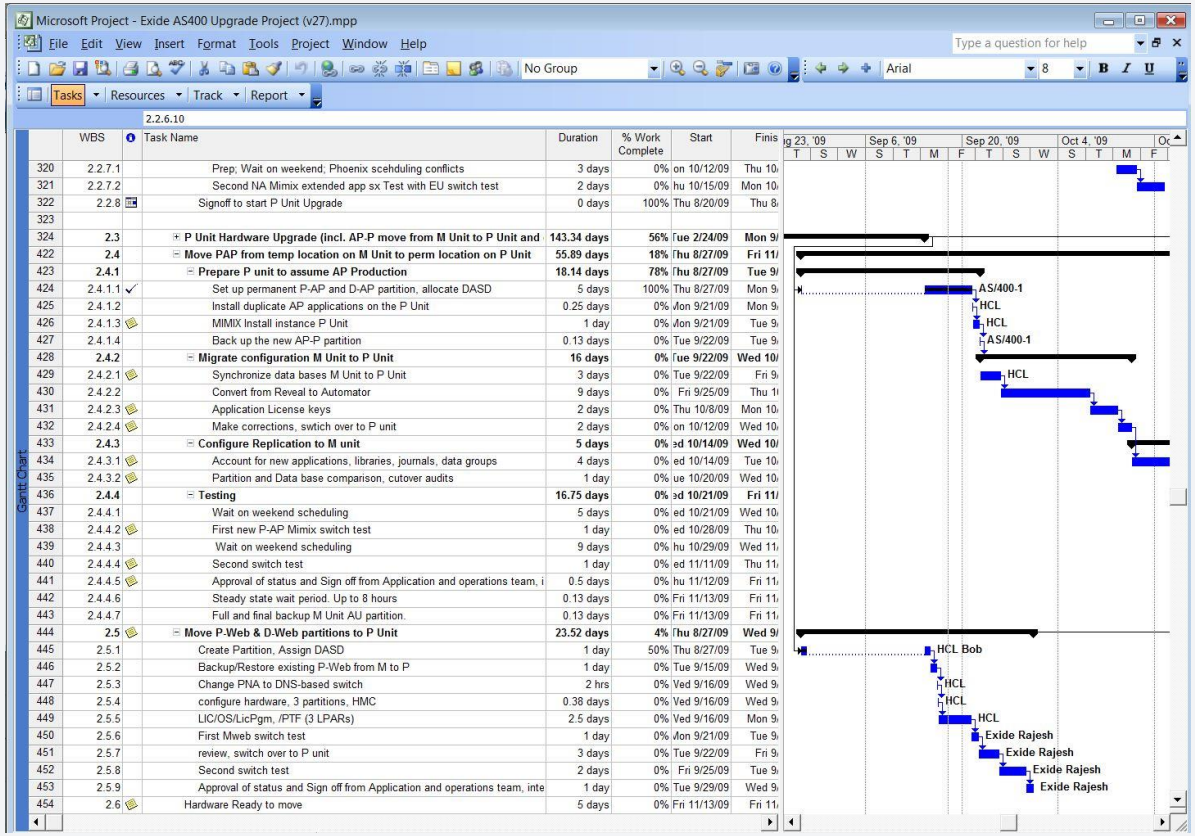
Project Management

- Software projects also have diverse stakeholders with competing agendas, which adds to the complexity of managing people.
- SE is thus as much a branch of the social sciences as it is of engineering.

Project Management – Non-SW Domain



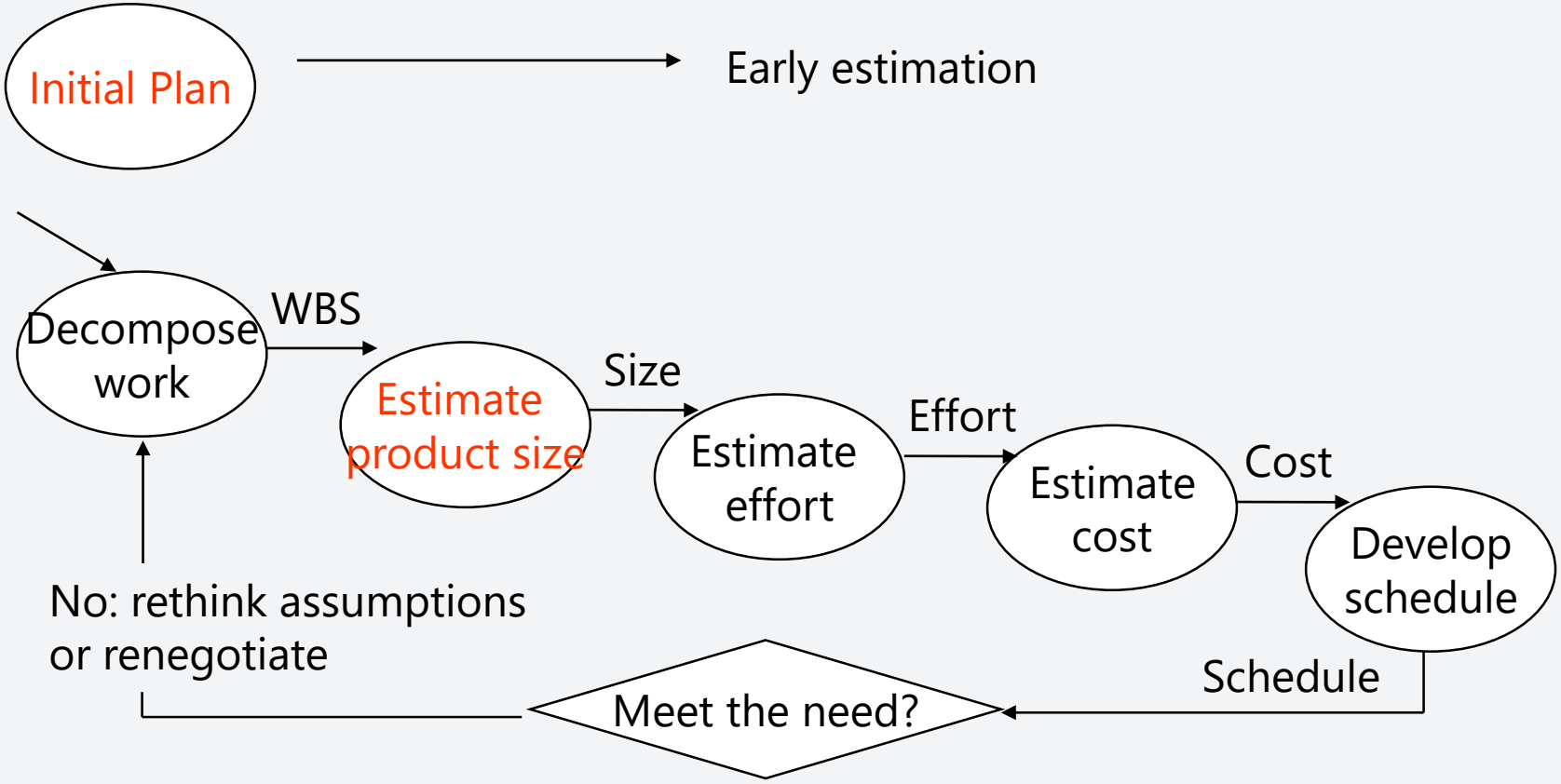
Project Management



Project Management - Gantt chart

- A **Gantt chart** is a type of bar **chart** that illustrates a project schedule.
- This **chart** lists
  - the tasks to be performed on the vertical axis
  - the time intervals on the horizontal axis.

The Planning Cycle



Hardware Productivity (verimlilik)

$$\frac{1{,}82 \text{ bicycles}}{\text{man. hour}}$$

$$\frac{3700 \text{ part} - X}{\text{man. year}}$$

P(roduction) M can make an accurate plan

# of units produced / Effort (Classical Manufacturing)

Software Productivity (verimlilik)

of units produced / Effort (Classical Manufacturing)

Which solution to a single software problem is more productive than other(s)?

There may be many solutions with different attributes.

- A solution execute more efficiently
- A solution which is more readable or easier to maintain
- Another solution ...

Software Productivity

- Is it meaningful to compare solutions with different attributes?
- Really not...
- Although it is not meaningful, PMs need productivity estimates to help
  - define the project cost, schedule
  - make decisions
  - improve software processes
  - and etc...

Productivity Estimates (Kestirim)

$$\text{Productivity} = \frac{\text{Attributes of Software}}{\sum \text{Effort}}$$

- Generally two types of metrics
  - Size-related metrics
  - Function-related metrics

Productivity Metrics

- Size-related metrics :
  - # of lines of delivered source code (e.g., LOC/man month)
  - # of object code instructions
  - # of pages of software documentation
- Function-related metrics: overall functionality of the delivered product.
  - **Ex:** Function Points, Object Points, Use Case Points etc.
    - (e.g., FP/man month)

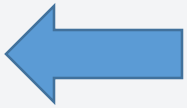
Size-related Metrics

- Productivity = # of LOCs / Effort
- The more expressive the PL, the lower the apparent productivity. ☹
  - But, may be resolved by using coefficients ☺
- Development time??
  - LOC → All software development activities?
  - However, LOC metric → Just programming activity
- Size-related Metrics CANNOT be obtained before the completion of the project (**Early size estimation**!!!)

LOC Conversion Ratios

From\To	Basic	Ada	Pascal	Cobol	Fortran	Jovial	C	Assembly
Basic	1.0	1.11	1.44	1.67	1.67	1.67	2.0	5.0
Ada	0.9	1.0	1.3	1.5	1.5	1.5	1.8	4.5
Pascal	0.69	0.77	1.0	1.15	1.15	1.15	1.38	3.5
Cobol	0.6	0.67	0.87	1.0	1.0	1.0	1.2	3.0
Fortran	0.6	0.67	0.87	1.0	1.0	1.0	1.2	3.0
Jovial	0.6	0.67	0.87	1.0	1.0	1.0	1.2	3.0
C	0.5	0.55	0.72	0.83	0.83	0.83	1.0	2.5
Assembly	0.2	0.22	0.29	0.33	0.33	0.33	0.4	1.0

Function-related metrics

- Functionality is independent of implementation language
- Productivity = # of FPs / Effort 
- FP is not a single characteristics but is computed by combining several different measurements or estimates.
- The best known is the function-point count by Albrecht and Gaffney in 1983
- IFPUG→ International Function Point Users Group

IFPUG Counting Method

- Assume that we have Functional User Requirements (After the completion of SRS document)
- Elementary Processes (Base Functional Component) are extracted from SRS

Source: Function Point Counting Practices Manual, Release 4.1

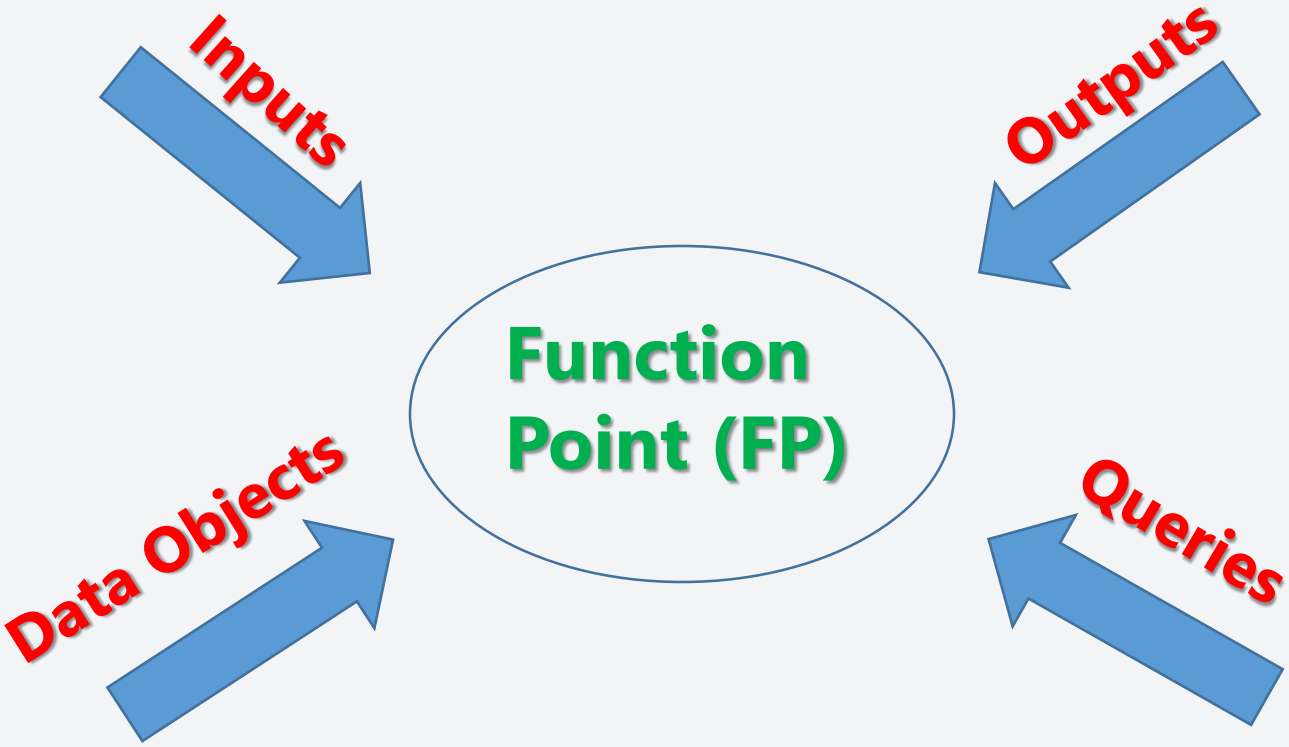
Examples of various COTS, SaaS, and open-source apps sized via pattern matching

Application	Size in Function Points
SAP	296,704
Windows 7	165,245
Office 2010	93,498
Skype	21,202
Apple iPhone	19,366
Linux	17,505
Google Docs	47,668
Google search engine	18,640
GPS navigation	1,508
Laser printer driver	1,248
Cochlear implant	1,041
Atomic watch	933

Source: IFPUG - The IFPUG Guide to IT and Software Measurement (2012)

FP Counting

- External Inputs(EI)
- External Outputs(EO)

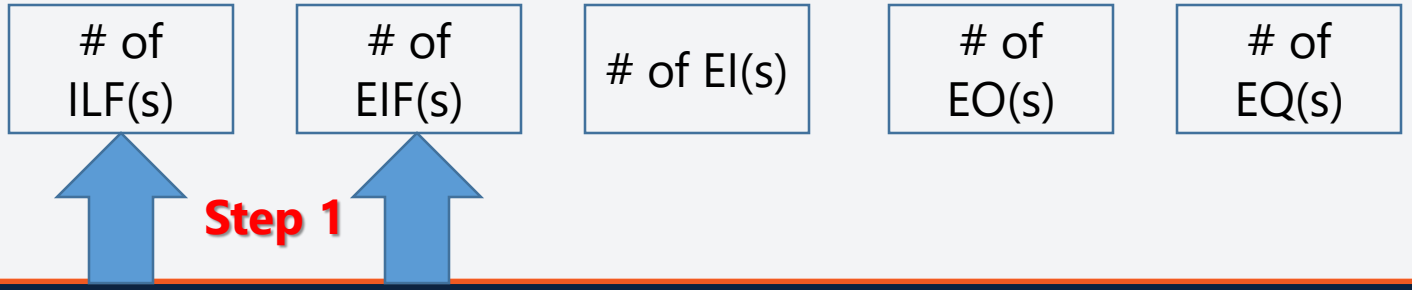


- Internal Logical Files (ILF)
- External Interface Files (EIF)
- External Inquiries (EQ)

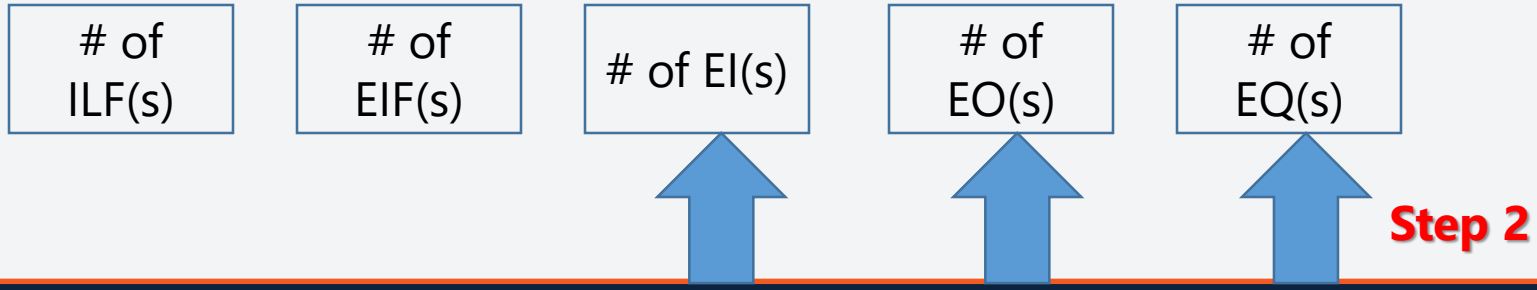
FP Counting Summary

# of ILF(s)	# of EIF(s)	# of EI(s)	# of EO(s)	# of EQ(s)

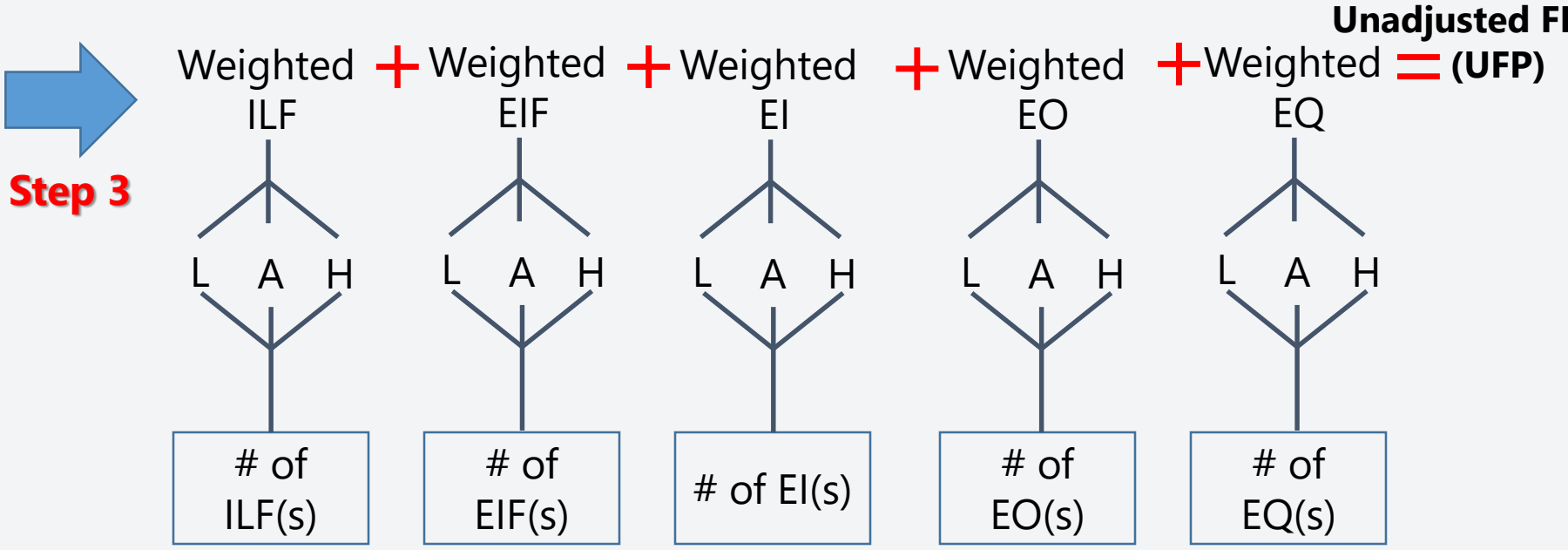
FP Counting Summary



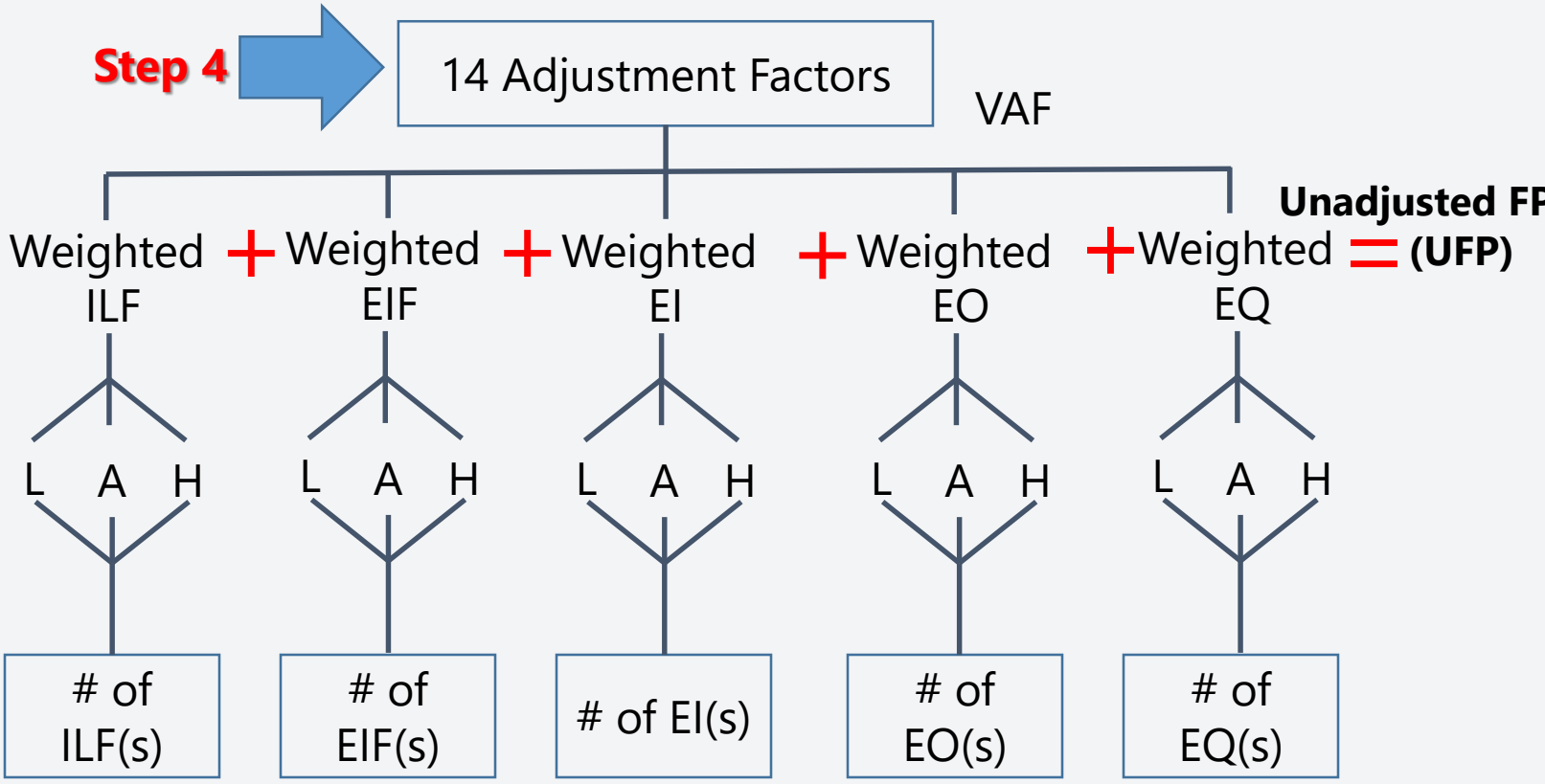
FP Counting Summary



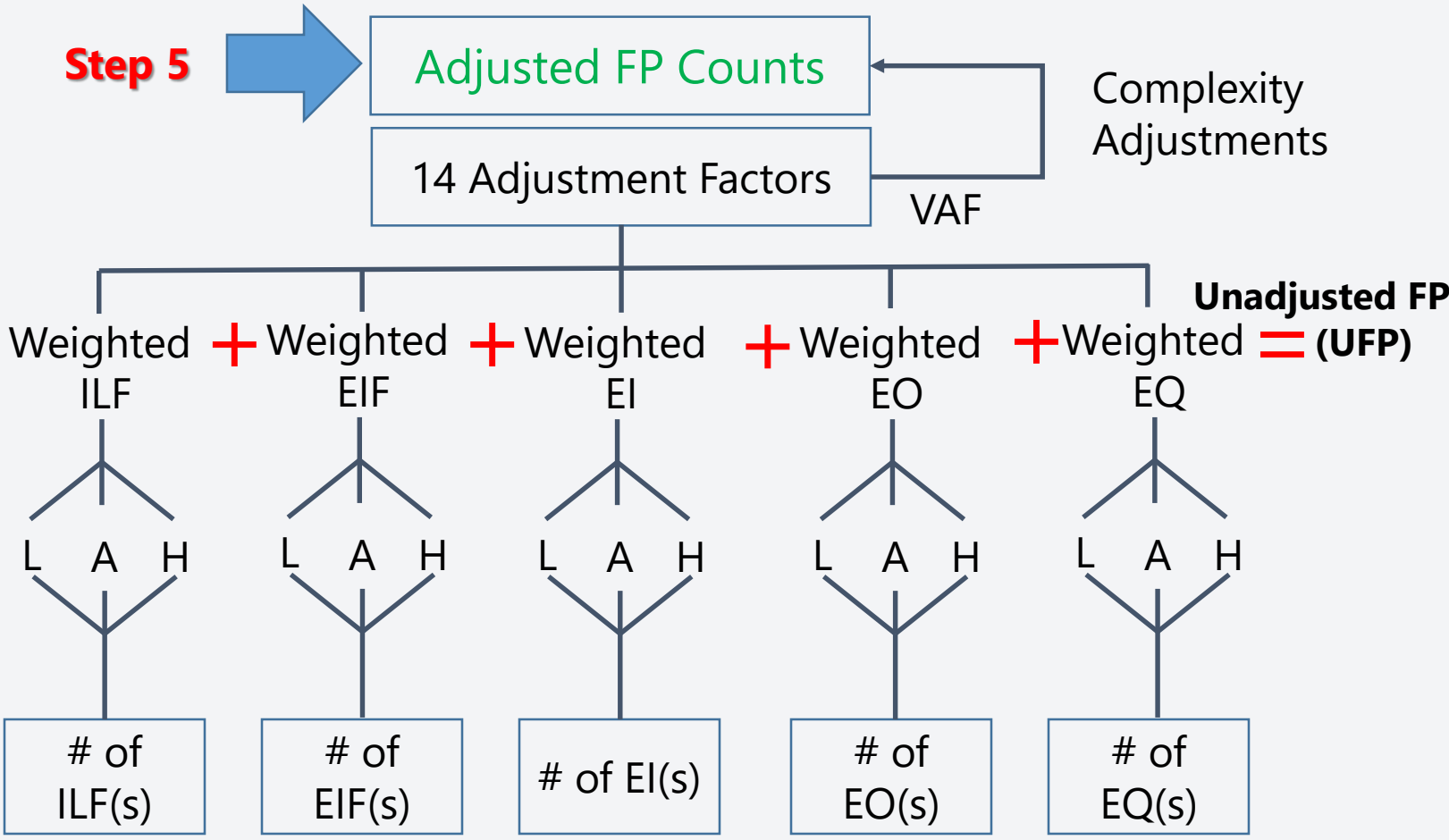
FP Counting Summary



FP Counting Summary



FP Counting Summary



Elementary Processes (Base Functional Component)

- **Elementary Process :** An elementary process is the smallest unit of activity that is meaningful to **the user(s)**.

- **Ex:** a user requires the ability to add a new employee to the application. The user definition of employee includes salary & dependent information. From the user perspective, the smallest unit of activity is to add a new employee.
  - Adding one of the pieces of information, such as salary or dependent, is NOT an activity that would qualify as an elementary process.

- The elementary process must be self-contained and leave the business of the application being counted in a consistent state.

Elementary Process Functions

- External Input
- External Output
- External Inquiry

} Transactional Functions

- Internal Logical File
- External Interface File

} Data Functions

IMPORTANT: Do **not** let the **name of the functions** confuse your calculations.

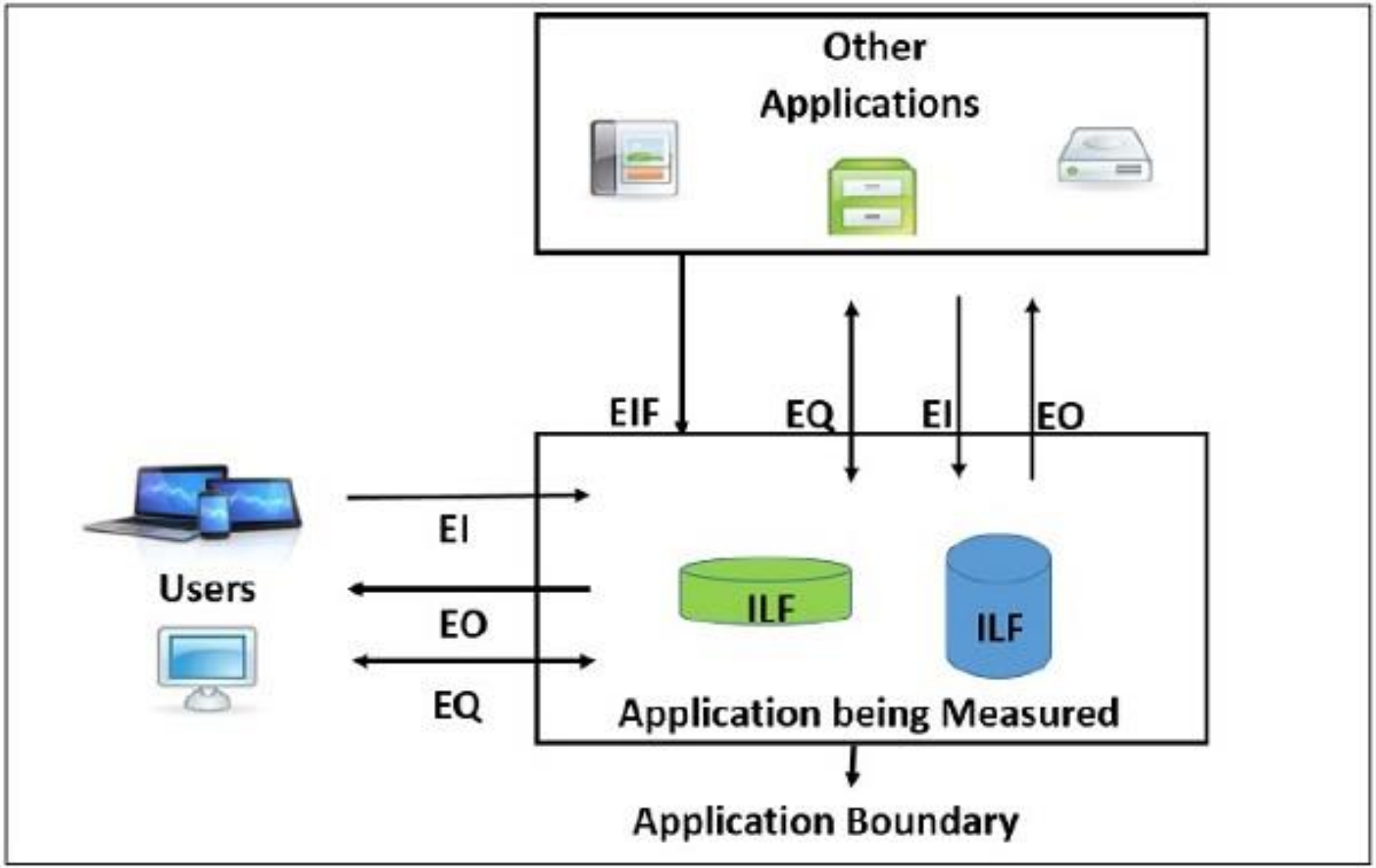
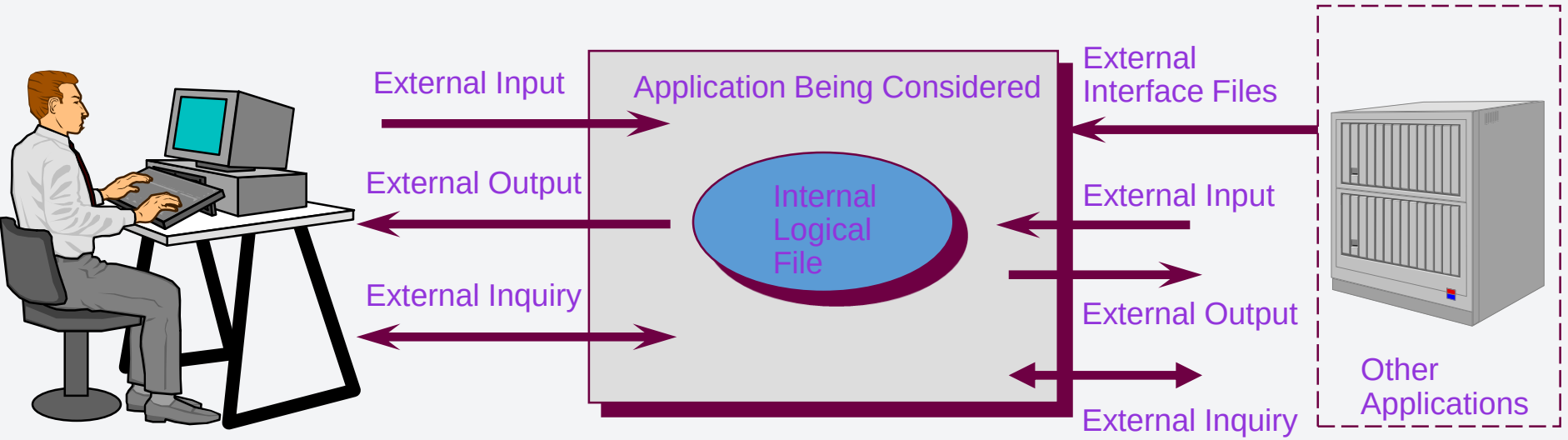


Figure 1: Application Boundary, Data Functions, Transaction Functions

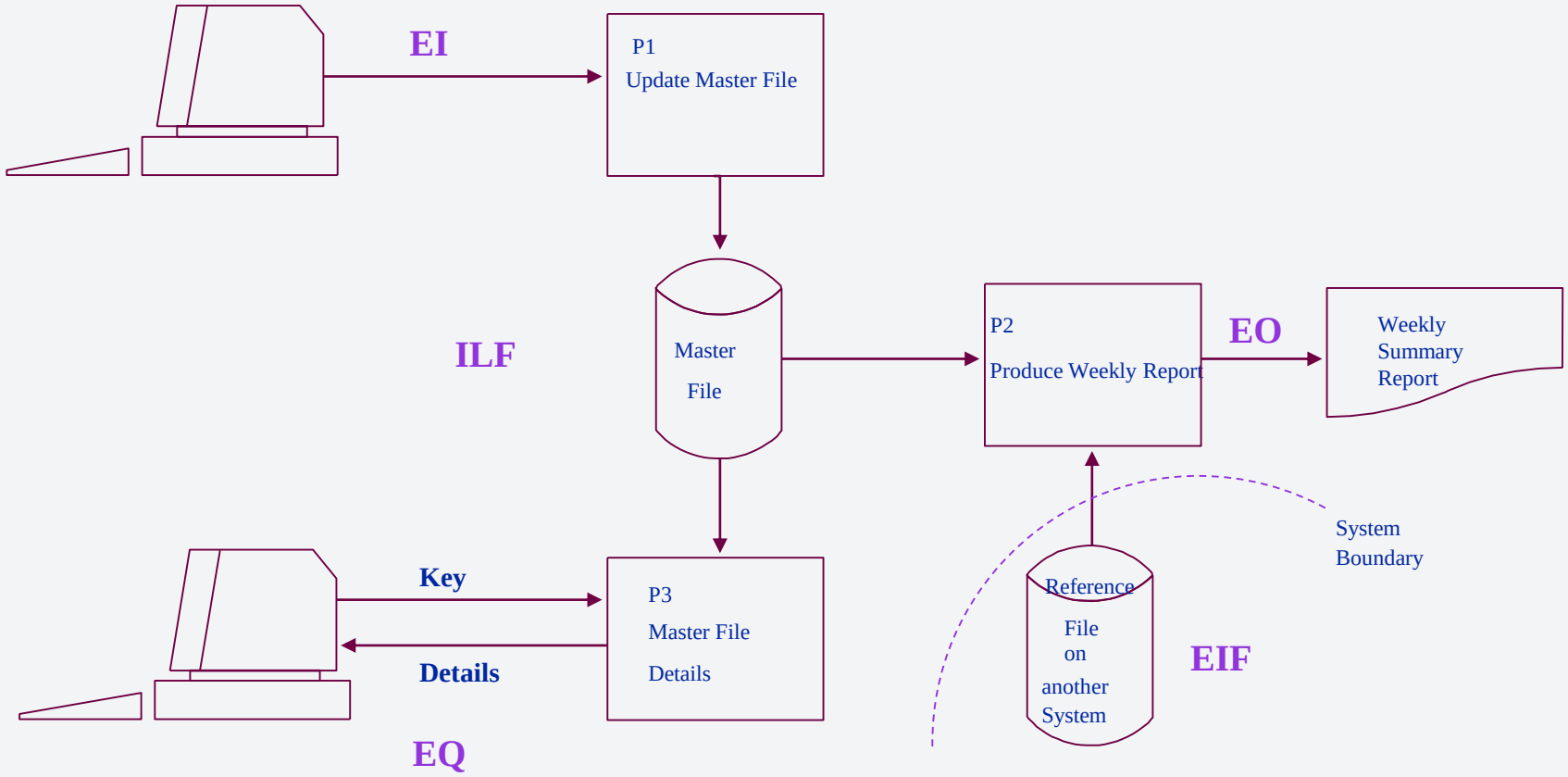
Function Points are a Unit of Measure



- Functionality as viewed from the user's perspective

Source: IFPUG In a Box https://www.ifpug.org/publications-products/ifpug-in-a-box/

FP Overview: What Is Counted



Source: IFPUG In a Box https://www.ifpug.org/publications-products/ifpug-in-a-box/

External Input (EI)

- An external input (EI) is an elementary process that processes data or control information that comes from outside the application's boundary.
- The primary intent of an EI is to maintain one or more ILFs and/or to alter the behavior of the system.

External Input (EI)

- Data may come from a data input screen or another application.
- An EI is how an application gets information.
- Data can be either control information or business information.
- Data may be used to maintain one or more Internal Logical Files.
- If the data is control information, it does not have to update an Internal Logical File.

External Output (EO)

- An external output (EO) is an elementary process that sends data or control information outside the application's boundary.
- The primary intent of an external output is to present information to a user through processing logic other than or in addition to the retrieval of data or control information.
  - The processing logic must contain at least one mathematical formula or calculation, or create derived data.
- An external output may also maintain one or more ILFs and/or alter the behavior of the system.

External Output (EO)

E.g.:

- the generated response in terms of log reports or files. This report or files are used as an input for other application.
- The output screen or reports are examples of EO.

External InQuiry (EQ)

- An external inquiry (EQ) is an elementary process that sends data or control information outside the application boundary.
- The primary intent of an external inquiry is to present information to a user through the retrieval of data or control information.
- The processing logic contains no mathematical formula or calculation, and creates no derived data.
- No ILF is maintained during the processing, nor is the behavior of the system altered.

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

- Internal Logical Files (ILF) is a user identifiable group of logically related data or control information maintained within the boundary of the application.
  - The primary intent of an ILF is to hold data maintained through one or more elementary processes of the application being counted.
- External Interface Files (EIF) is a user identifiable group of logically related data or control information referenced by the application, but maintained within the boundary of another application.
  - The primary intent of an EIF is to hold data referenced through one or more elementary processes within the boundary of the application counted. This means an EIF counted for an application must be in an ILF in another application.

Definitions

- A *data element type* (DET) is a unique user recognizable, **non-repeated** field.
- A *record element type* (RET) is a user recognizable **subgroup** of data elements within an ILF or EIF.
- A *file type referenced* (FTR) is
  - An ILF read or maintained by a transactional function or
  - An EIF read by a transactional function

Functional Complexity of **EIF & ILF**

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

Translation of ILFs' complexity to unadjusted FP

Functional Complexity Rating	Unadjusted Function Points
Low	7
Average	10
High	15

Translation of EIFs' complexity to unadjusted FP

Functional Complexity Rating	Unadjusted Function Points
Low	5
Average	7
High	10

Functional Complexity for External Inputs (EI)

	1-4 DET	5-15 DET	16 or more DET
0 – 1 FTR	Low	Low	Average
2 FTR	Low	Average	High
3 or more FTR	Average	High	High

Functional Complexity for External Outputs (EO) and External Inquiries (EQ)

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

Translation Complexity of External Inputs (EI) and External Inquiries (EQ) to UFP

Functional Complexity Rating	Unadjusted Function Points
Low	3
Average	4
High	6

Translation Complexity of External Outputs (EO) to UFP

Functional Complexity Rating	Unadjusted Function Points
Low	4
Average	5
High	7

IFPUG's Unadjusted Function Point Table

	Function Levels			
Components	Low	Average	High	Total
External Inputs	x 3	x 4	x 6	
External Outputs	x 4	x 5	x 7	
External Inquiries	x 3	x 4	x 6	
Internal Logical Files	x 7	x 10	x 15	
External Interface Files	x 5	x 7	x 10	
			Total Unadjusted FP (UFP):	

Total Degree of Influence (TDI)

Characteristic	Degree of Influence (DI) (0-5)
1. Data communication	
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):	**Min →0 Max →70**

Calculating Adjusted Function Points

- To convert the UAF to Adjusted Function Points the Value Adjustment Factor (VAF) is calculated as follows:
  - 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 the TDI into the following equation to produce the value adjustment factor.

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

- For example, the following VAF is calculated if TDI is 42.

  VAF = (42 * 0.01) + 0.65 ➔ 1.07

Calculating Adjusted Function Points

- Use the following formula to calculate the development project function point count.

$$\mathbf{DFP = (UFP + } \textit{CFP}\mathbf{) * VAF}$$

- Where:
  - DFP is the development project function point count
  - UFP is the unadjusted function point count for the functions that will be available after installation
  - CF*P is the unadjusted function points added by the conversion unadjusted function point count*
  - VAF is the value adjustment factor

Calculating Adjusted Function Points

- Example:
  - Using the complexity and contribution counts, UFP is calculated as 115 and CFP as 3. The value adjustment factor (VAF) for this example is 1.05.
- DFP = (UFP + CFP) * VAF
- DFP = (115 + 3) * 1.05
- **DFP = 123.9** or **124**

Function Types - Complexity Assessments

	Data Function Types		Transaction Function Types		
	Internal Logical Files (ILF)	External Interface Files (EIF)	External Input (EI)	External Output (EO)	External Inquiry (EQ)
Elements evaluated for technical complexity assessment	**REcord Types (RET)**: User recognizable sub groups of data elements within an ILF or an EIF. It is best to look at logical groupings of data to help identify them.		**File Type Referenced (FTR):** File type referenced by a transaction. An FTR must be an Internal Logical File (ILF) or External Interface File (EIF).		
	Data Element Types (DET): A unique user recognizable, non-recursive (non-repetitive) field containing dynamic information. If a DET is recursive then only the first occurrence of the DET is considered not every occurrence.				

Source: SENG421: Software Metrics (3rd year, Software Engineering Course)
http://people.ucalgary.ca/~far/Lectures/SENG421/index.html

Example

External Interface File (EIF)

EIF: Employee Administration (DB)

Employee
- Name
- ID
- Birth date
- Payment Reference

RET # 1

Weekly Payment
- Hourly Rate
- Payment Office

RET # 2

Monthly Payment
- Salary Level

• PAYROLL Software

Source: SENG421: Software Metrics (3rd year, Software Engineering Course) http://people.ucalgary.ca/~far/Lectures/SENG421/index.html

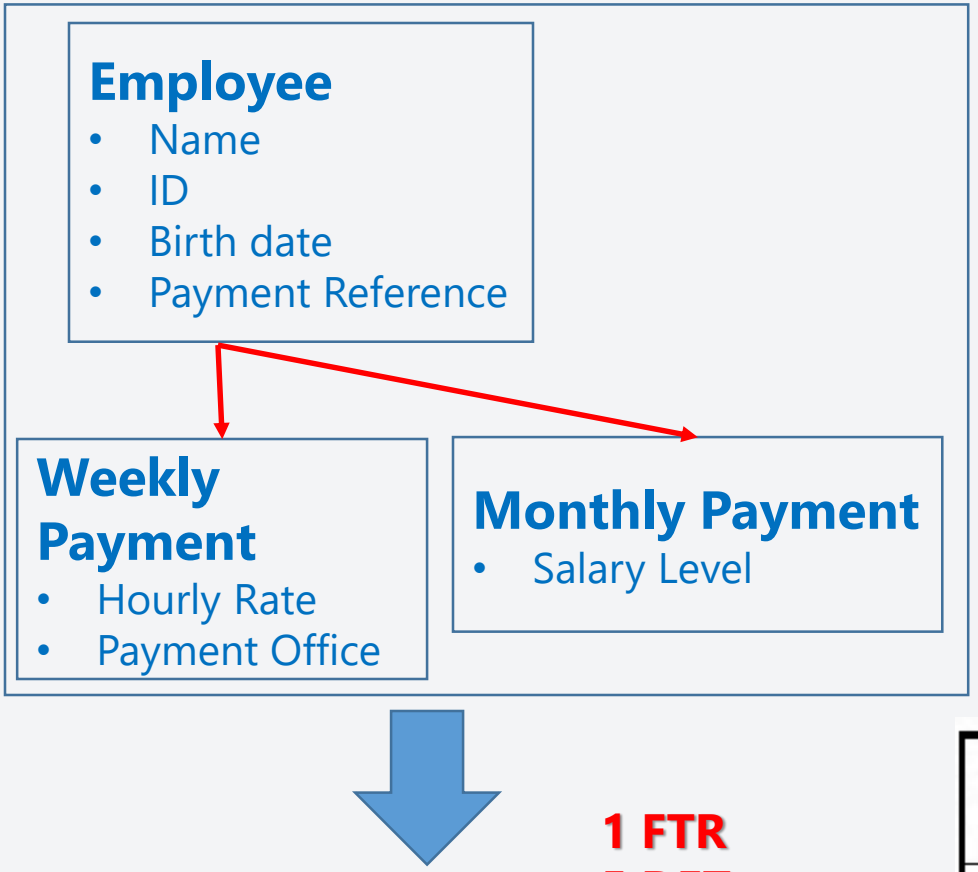
2 RET
7 DET

EIF		#DET		
		1-19	20-50	> 50
#RET	1	low (5)	low (5)	average (7)
	2-5	low (5)	average (7)	high (10)
	> 5	average (7)	high (10)	high (10)

Example

External Input (EI)

ILF: Employee Administration (DB)



1 FTR
5 DET

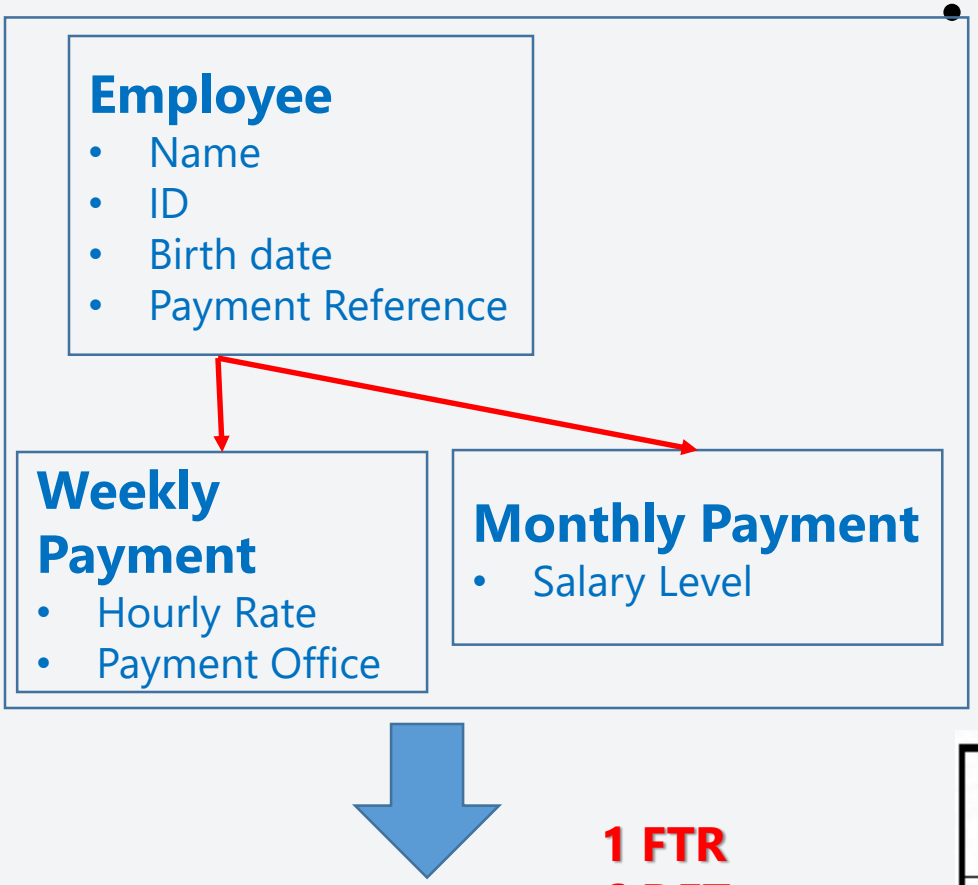
- Enter a new employee with Monthly Payment (EI)
  - Name
  - ID
  - Birth date
  - Payment Reference
  - Salary Level

EI		#DET		
		1-4	5-15	> 15
#FTR	1	low (3)	low (3)	average (4)
	2	low (3)	average (4)	high (6)
	> 2	average (4)	high (6)	high (6)

Example

External Input (EI)

ILF: Employee Administration (DB)



1 FTR
6 DET

- Enter a new employee with Weekly Payment (EI)
  - Name
  - ID
  - Birth date
  - Payment Reference
  - Hourly Rate
  - Payment Office

EI		#DET		
		1-4	5-15	> 15
#FTR	1	low (3)	low (3)	average (4)
	2	low (3)	average (4)	high (6)
	> 2	average (4)	high (6)	high (6)

External Output(EO)

Example

ILF: Employee Administration (DB)

Employee
- Name
- ID
- Birth Date
- Payment Reference

<= 40 Years
Jerome Iginla, 10-02-1966
Rehet Warner, 05-03-1966
<= 45 Years
Chris Cheilos, 23-03-1961

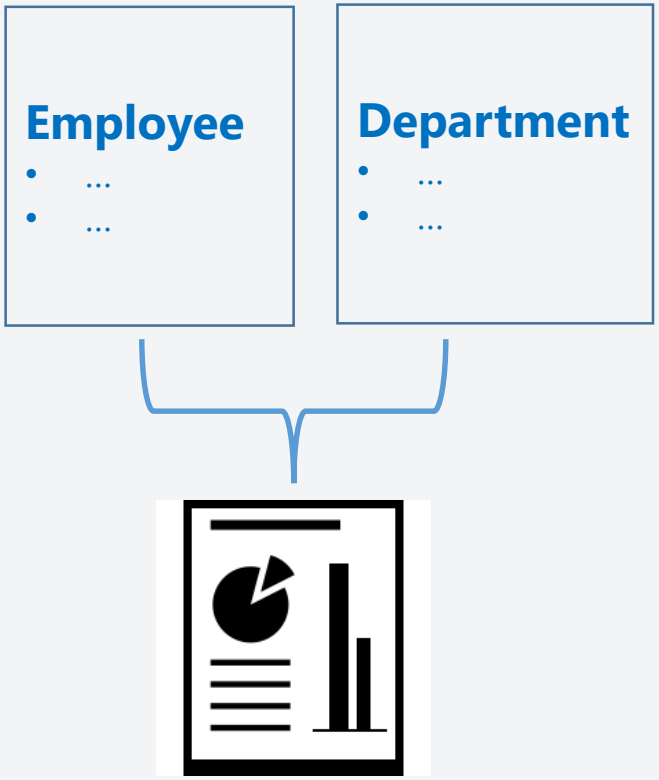
1 FTR
2 DET

- Report of all Employees containing Names and Birth Dates, sorted by age (EO)

EO		#DET		
		1-5	6-19	> 19
#FTR	1	low (4)	low (4)	average (5)
	2-3	low (4)	average (5)	high (7)
	> 3	average (5)	high (7)	high (7)

External Inquiries(EQ)

Example



Report of all Employees belonging to Department X containing Names, Birth Dates, and showing the Department Name (EQ)

- Files (ILF): Employee, Department
- 2 FTR: Employee, Department
- 3 DET: Name (Employee), Birth Date (Employee), Department Name (Department)

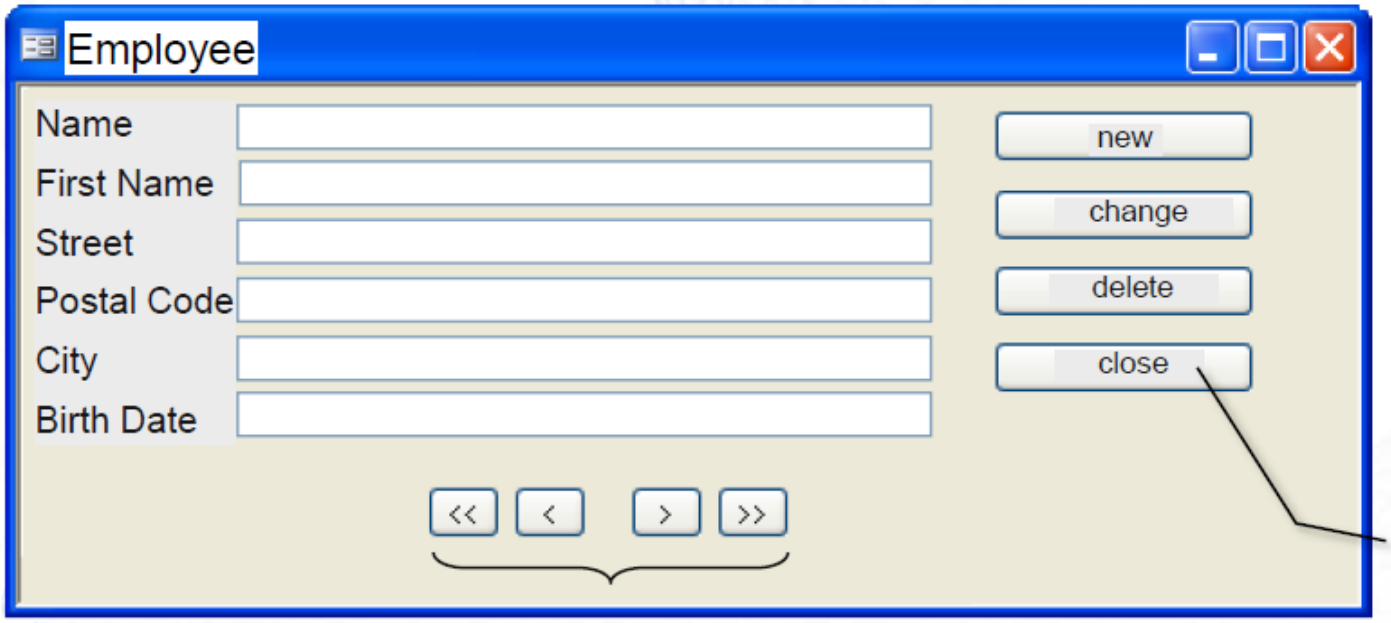
- **Report**

2 FTR
3 DET

EQ		#DET		
		1-5	6-19	> 19
#FTR	1	low (3)	low (3)	average (4)
	2-3	low (3)	average (4)	high (6)
	> 3	average (4)	high (6)	high (6)

FP Counting - GUI

Example



3 External Inputs

"close" button does not count, because it is not a transaction in its own right involving access to ILFs, EIFs

1 External Inquiry (input side)

```
External Input (new, 1 FTR, 7 DET, Low)          = 3 FP
External Input (change, 1 FTR, 7 DET, Low)       = 3 FP
External Input (delete, 1 FTR, 7 DET, Low)       = 3 FP
External Inquiry (navigate, 1 FTR, 7 DET, Low)   = 3 FP
```

12 FP

Example

ILF Counting Examples

- **Example: Audit Data for Inquiries and Reports**
- **User Requirements:** Analysis of the following user security requirements showed a need for audit data:

1. Allow or deny user access to each screen in the application.
2. Change a user's access to each screen.
3. Report on any screen security added or changed using the following data:
   - Identification of the user who is adding or changing security information
   - The user and screen security that was added or changed
   - The user and screen security images before and after a change was made
   - Date and time the add or change occurred.
4. Capture audit data to monitor and report daily security activity. This requirement was determined when a design was implemented to satisfy the user's screen security requirements.

Source: Function Point Counting Practices Manual Release 4.3.1

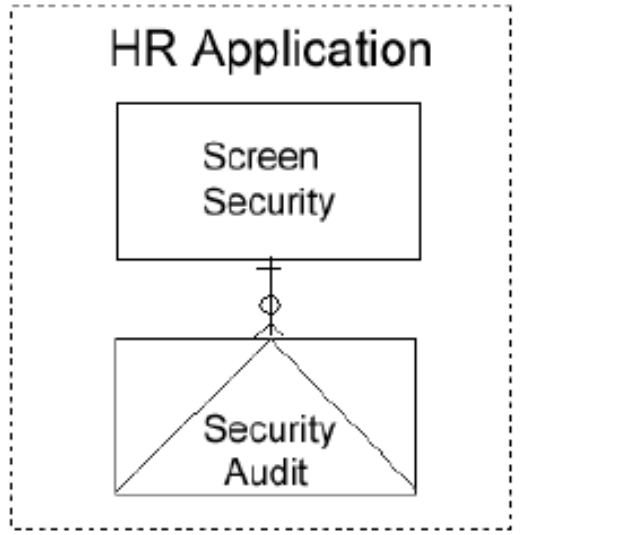
Example

ILF Cou

- **Example: A** **rts**
- **User Requ** user security requiremer

1. Allow or der
2. Change a us
3. Report on ar lowing data:
   - Identificatior
   - The user and
   - The user and de
   - Date and tim
4. Capture auc tivity. This requiremen nted to satisfy the user's sc

ERD



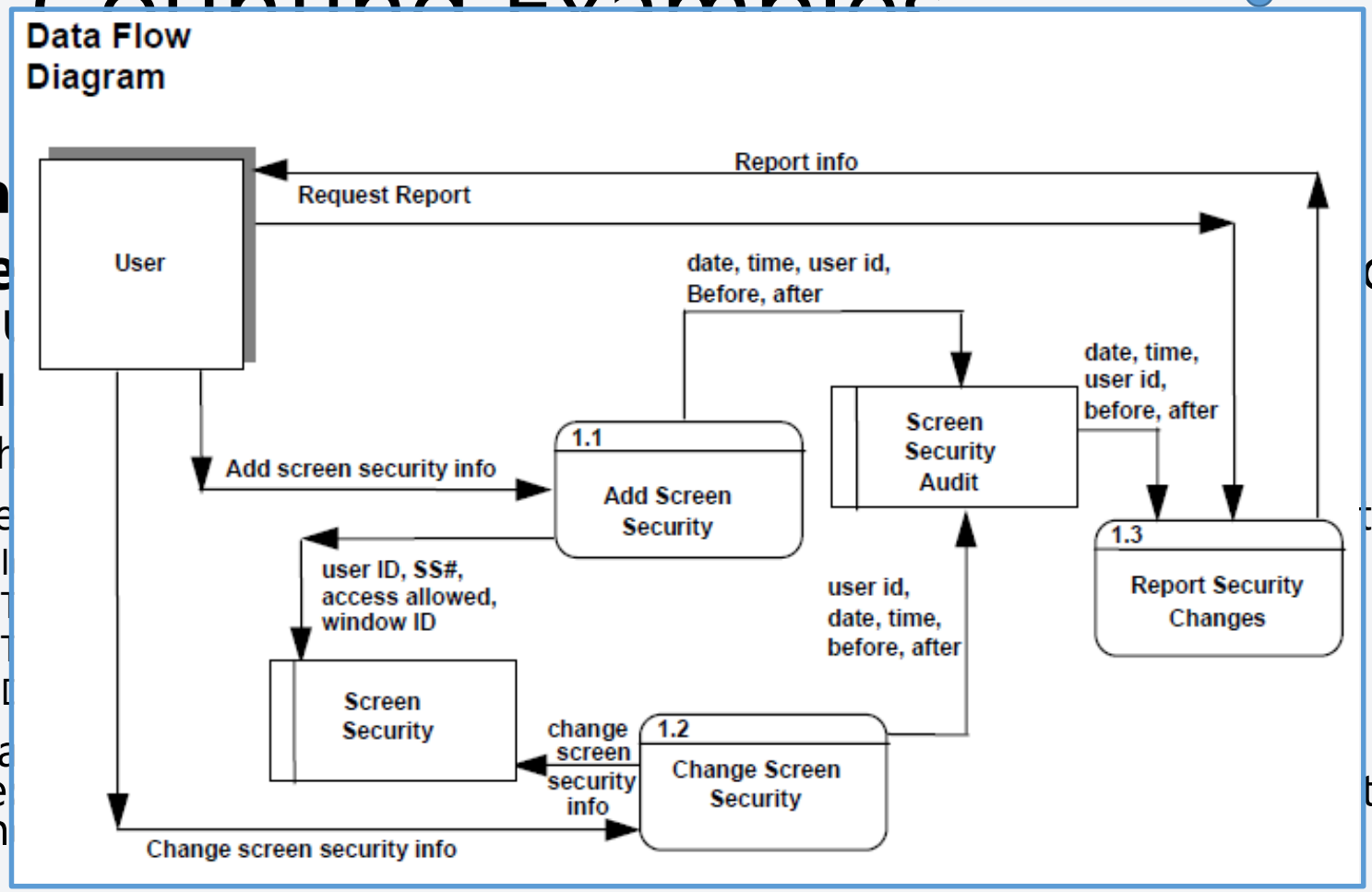
Source: Function Point Counting Practices Manual Release 4.3.1

Example

ILF Counting Examples

- **Exa**
- **Use** curity requ

1. Al
2. Ch
3. Re ta:
   - I
   - T
   - T
   - D
4. Ca tisfy re th



Source: Function Point Counting Practices Manual Release 4.3.1

Example

ILF Counting Examples

- Step 1
- **Identify Data Functions**
- Use the Data Function Identification rules to determine whether Screen Security Audit data is a data function. The following table shows the analysis for Screen Security Audit data.

Source: Function Point Counting Practices Manual Release 4.3.1

Example

ILF Counting Examples

Data Function Identification Rules	Does the Rule Apply?
1. Identify all logically related and user recognizable data or control information within the counting scope	Screen Security and Screen Security Audit.
2. Exclude entities that are not maintained by any application	There are no entities of this type.
3. Group related entities that are entity dependent	Screen Security and Screen Security Audit are related. Screen Security Audit is dependent upon Screen Security. They are grouped together into a single data function.
4. Exclude those entities referred to as code data	There are no entities of this type.
5. Exclude entities that do not contain attributes required by the user	There are no entities of this type.
6. Remove associative entities that contain additional attributes not required by the user and associative entities that contain only foreign keys; group foreign key attributes with the primary entities	There are no entities of this type.

Source: Function Point Counting Practices Manual Release 4.3.1

Example

ILF Counting Examples

- Step 1
- The Screen Security Audit is not counted as a data function on its own because it is dependent upon Screen Security. Screen Security Audit is part of the Screen Security data function.

Example

ILF Counting Examples

- Step 2
- **Classify Data Functions**
- The following table shows the analysis to determine whether the Screen Security information is classified as an ILF.

Data Function Classification Rules	Does the Rule Apply?
1. Classify as an ILF if the data is maintained by the application being measured	The Screen Security data function is maintained within the application.
2. Classify as an EIF, if it: • Is referenced, but not maintained, by the application being measured and • Is identified in an ILF in one or more other applications	Classified as an ILF; consequently, no EIFs are identified.

Example

ILF Counting Examples

- Step 2
- **Classify Data Functions**
- Based on the analysis, the Screen Security information is classified as an ILF.

Example

ILF Counting Examples

- **Step 3 Count DETs**
- **For DETs**, look at each attribute associated with the Screen Security ILF and determine whether the DET counting rules apply.
- The Screen Security ILF includes:
  - User ID
  - SS #
  - Window ID
  - Access Allowed
  - Date Change Made
  - Time Change Made
  - Before Image
    - User ID Before
    - Window ID Before
    - Access Allowed Before
  - • After Image
    - User ID After
    - Window ID After
    - Access Allowed After

Source: Function Point Counting Practices Manual Release 4.3.1

Example

ILF Counting Examples

- **Step 3 Count DETs**
- **For DETs**, look at each attribute associated with the Screen Security ILF and determine whether the DET counting rules apply.

Data Function DET Counting Rules	Does the Rule Apply?
1. Count one DET for each unique user recognizable, non-repeated attribute maintained in or retrieved from the data function through the execution of all elementary processes within the counting scope	User ID, SS #, Window ID, Access Allowed, Date Change Made and Time Change Made.
2. Count only those DETs being used by the application being measured when two or more applications maintain and/or reference the same data function	There are no attributes of this type.
3. Count one DET for each attribute required by the user to establish a relationship with another data function	There are no attributes of this type.
4. Review related attributes to determine if they are grouped and counted as a single DET or whether they are counted as multiple DETs; grouping will depend on how the elementary processes use the attributes within the application	User ID Before, Window ID Before and Access Allowed Before are grouped and counted as Before Image. The same is also done for the After Image attributes.

Example

ILF Counting Examples

- **Step 4 Count RETs**
- **For RETs**, identify subgroups based on the RET counting rules.

RET Counting Rules	Does the Rule Apply?
1. Count one RET for each data function (i.e., by default, each data function has one sub-group of DETs to be counted as one RET)	Count one RET for the Screen Security ILF.
2. Count one additional RET for each of the following additional logical sub-groups of DETs (within the data function) that contains more than one DET:	
• associative entity with non-key attributes	There are no entities of this type.
• sub-type (other than the first sub-type) and	There are no entities of this type.
• attributive entity, in a relationship other than mandatory 1-1	Screen Security Audit is an attributive entity in an optional 1-M relationship. Count an additional RET for Screen Security Audit.

Example

ILF Counting Examples

The RET and DET totals for Screen Security are shown in the following table.

RETs		DETs	
• Screen Security • Screen Security Audit		• User ID • SS # • Window ID • Access Allowed • Date Change Made • Time Change Made • Before Image • After Image	
Total	2 RETs	**Total**	8 DETs

Step 5 **Determine Functional Complexity**

2 RETs and 8 DETs	Complexity is Low

Step 6 **Determine Functional Size**

Functional Size of 1 Low ILF	7 FP

Source: Function Point Counting Practices Manual Release 4.3.1

FP Conversion Factors

Language	LOC / FP
Assembler	320
Cobol, Fortran	106
Pascal	91
RPG, PLI	80
ADA	71
Lisp, Basic	64
4th Generation DB	40
Java	30
Smalltalk	21

New Tables

FP Conversion Factors

Language	QSM SLOC/FP Data			
	Avg	Median	Low	High
ABAP (SAP) *	28	18	16	60
ASP*	51	54	15	69
Assembler *	119	98	25	320
Brio +	14	14	13	16
C *	97	99	39	333
C++ *	50	53	25	80
C# *	54	59	29	70
COBOL *	61	55	23	297
Cognos Impromptu Scripts +	47	42	30	100
Cross System Products (CSP) +	20	18	10	38
Cool:Gen/IEF *	32	24	10	82
Datastage	71	65	31	157

Source: https://www.qsm.com/resources/function-point-languages-table

New Tables

FP Conversion Factors

Language	QSM SLOC/FP Data			
	Avg	Median	Low	High
Excel *	209	191	131	315
Focus *	43	45	45	45
FoxPro	36	35	34	38
HTML *	34	40	14	48
J2EE *	46	49	15	67
Java *	53	53	14	134
JavaScript *	47	53	31	63
JCL *	62	48	25	221
LINC II	29	30	22	38
Lotus Notes *	23	21	19	40
Natural *	40	34	34	53
.NET *	57	60	53	60
Oracle *	37	40	17	60

Source: https://www.qsm.com/resources/function-point-languages-table

New Tables

FP Conversion Factors

Language	QSM SLOC/FP Data			
	Avg	Median	Low	High
PACBASE *	35	32	22	60
Perl *	24	15	15	60
PL/I *	64	80	16	80
PL/SQL *	37	35	13	60
Powerbuilder *	26	28	7	40
REXX *	77	80	50	80
Sabretalk *	70	66	45	109
SAS *	38	37	22	55
Siebel *	59	60	51	60
SLOGAN *	75	75	74	75
SQL *	21	21	13	37
VB.NET *	52	60	26	60
Visual Basic *	42	44	20	60

Source: https://www.qsm.com/resources/function-point-languages-table

References

- Function Point Counting Practices Manual Release 4.3.1