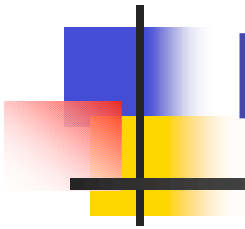
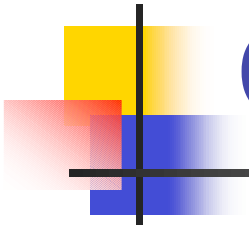
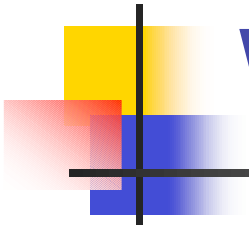


Function Points



Contents

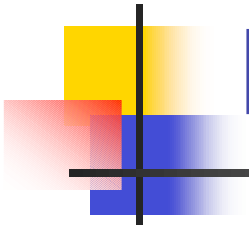
1. Introduction
2. Concept
3. Procedure
4. Summary



What is a Function Point

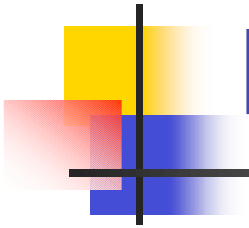
- A unit to measure functionality of Software
- A measure based on elementary processes* based on the users view

- * Elementary Process: the smallest unit of activity meaningful to the user



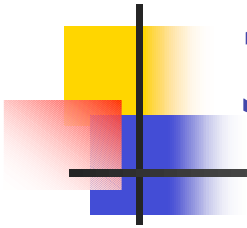
History

- 1975 IBM internal
- 1979 Albrecht
- 1982 DeMarco
- 1983 British Mark II
- 1984 IBM (revised)
- 1985 SPR
- 1987 IFPUG 1
- 1988 IFPUG 2
- 1990 IFPUG 3
- **1995 IFPUG 4**
- 1989 Texas Instruments
- 1997 ISO (working group)



Prerequisites

- user requirements
- e.g. ER-Diagram, Process Model, workflow diagrams



3. Procedure

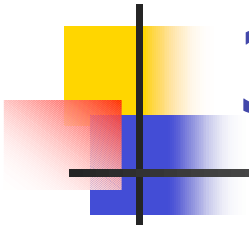
Determine Type of Count

Identify:
- Scope
- Application Boundary

Count:
- Data Functions
- Transactional Functions

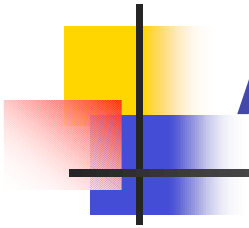
Determine:
- unadjusted FP Count
- Value Adjustment Factor

Calculate Adjusted FP Count



3.1 Counting Types

- Development
- Enhancement
- Application

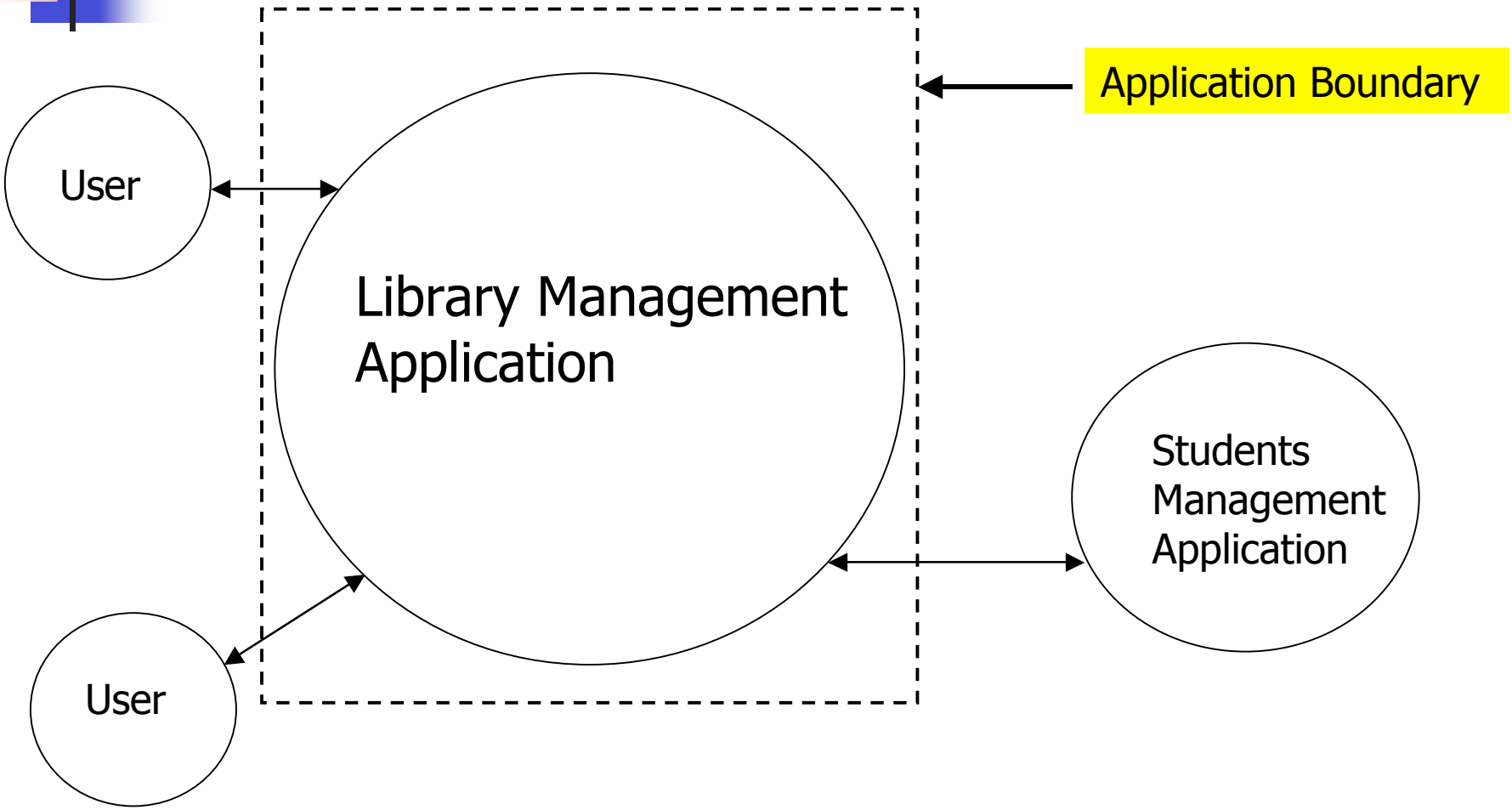


Application Boundary

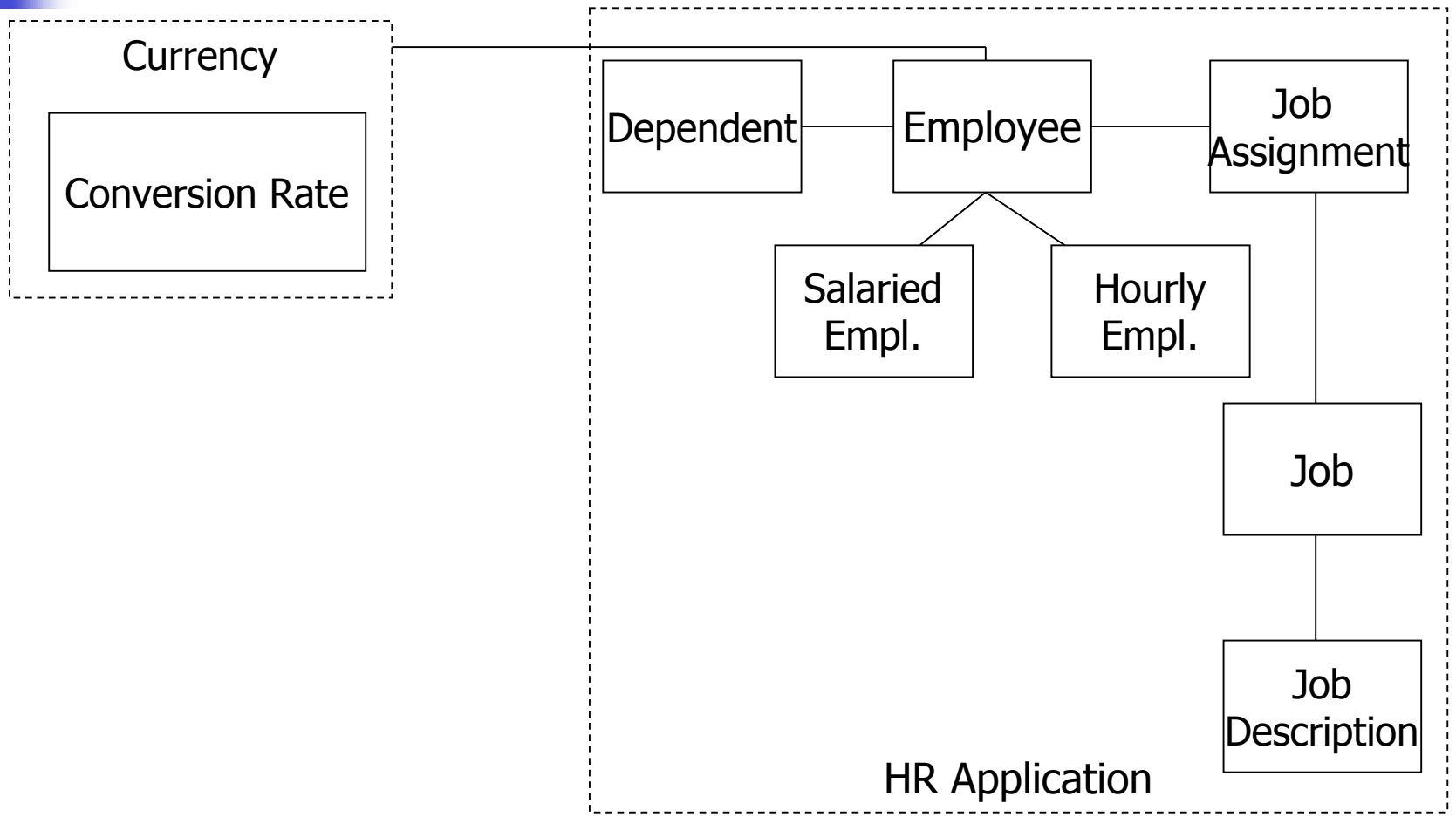
- Rule(s): based on User‘s view
- independent of technical ressources

- helps identifying EIFs
- Includes all that is done or held inside the application

Application Boundary



Application Boundary: ER-Diagram



Function Types

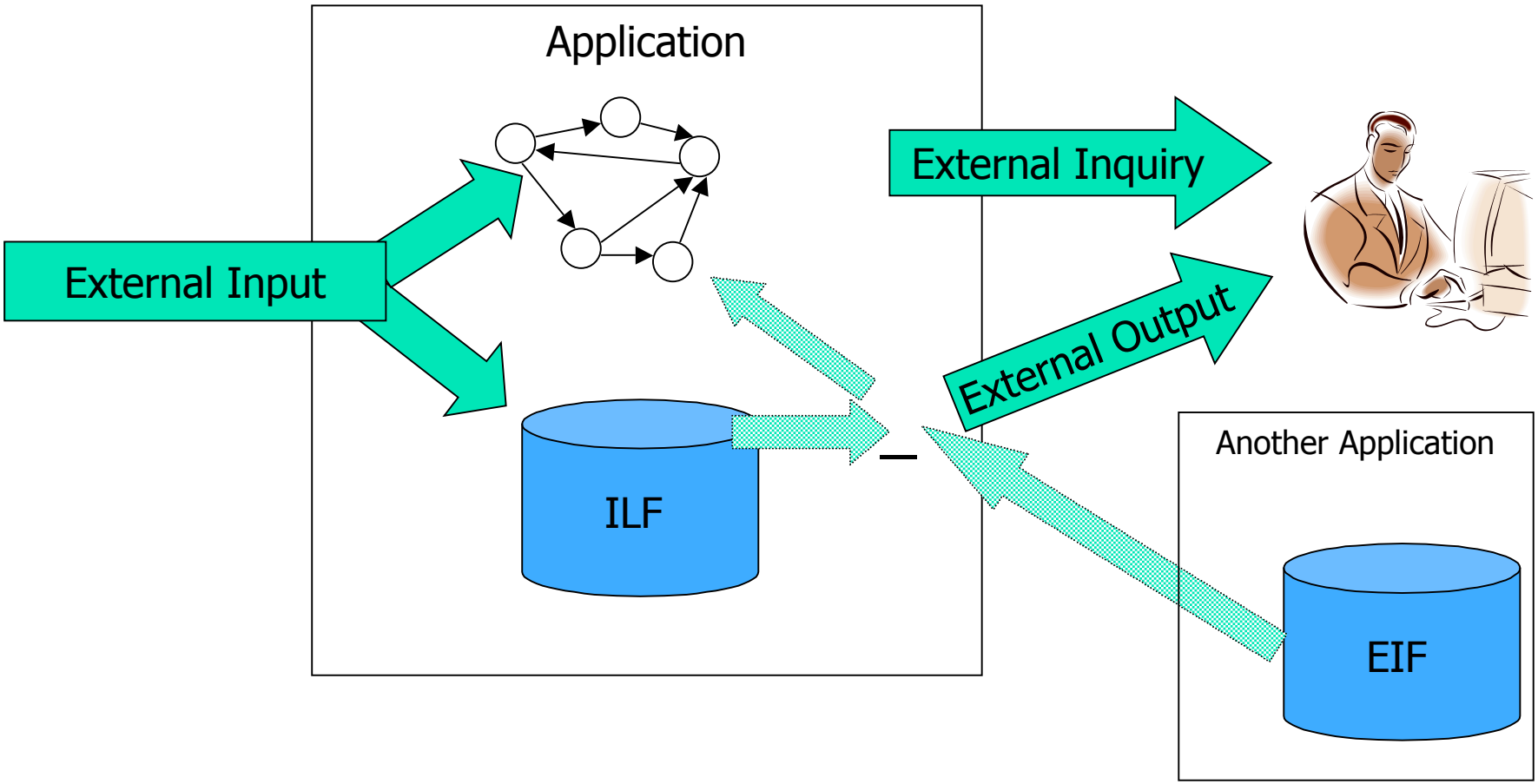
Data Functions:

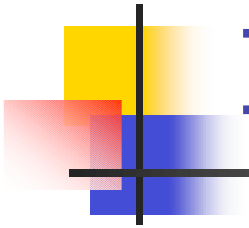
- Internal Logical File (ILF)
- External Interface File (EIF)

Transactional Functions:

- External Input (EI)
- External Outputs (EO)
- External Inquiry (EQ)

Function Types





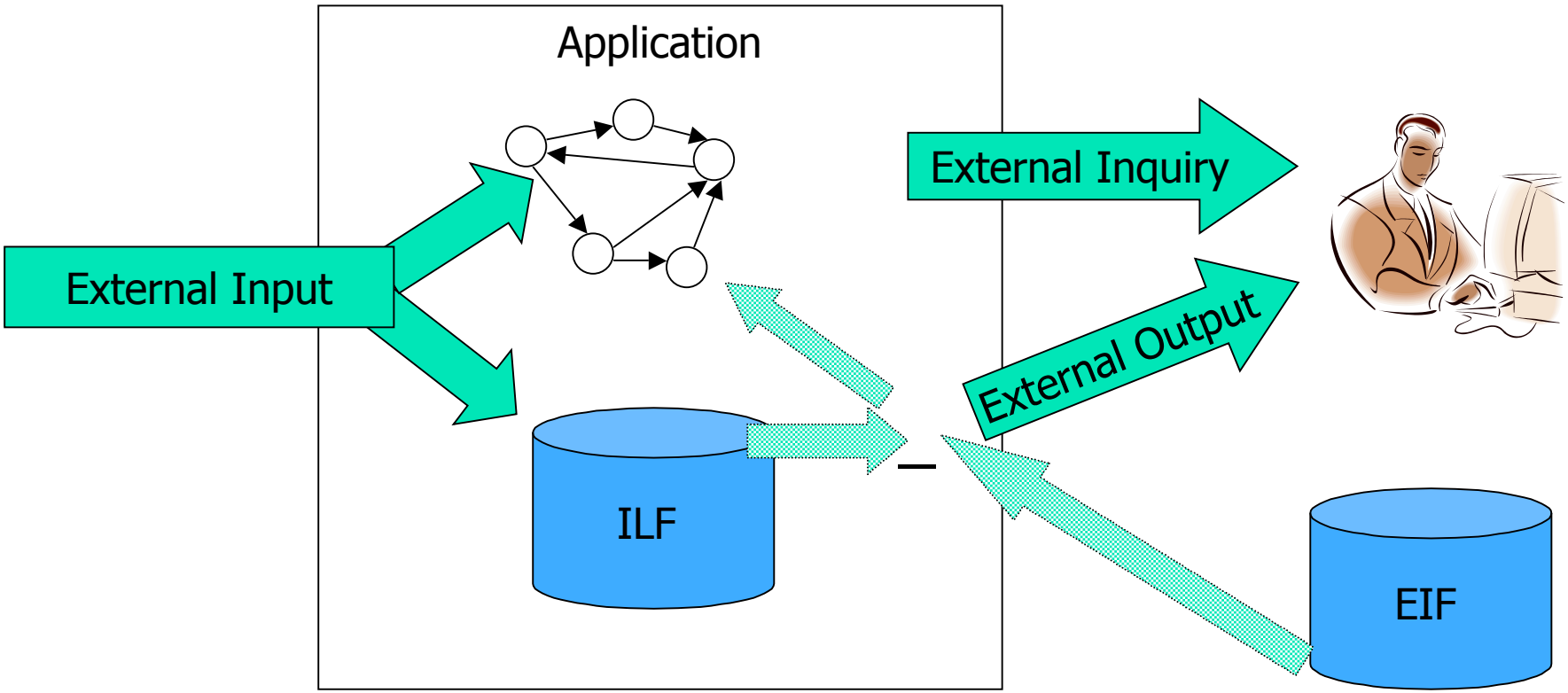
Internal Logical File

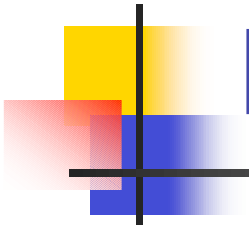
A ILF is a group of user identifiable, logically related information which is:

- Data or Control Information
- maintained within the boundary of the application

- Primary intent: hold data

Data Functions





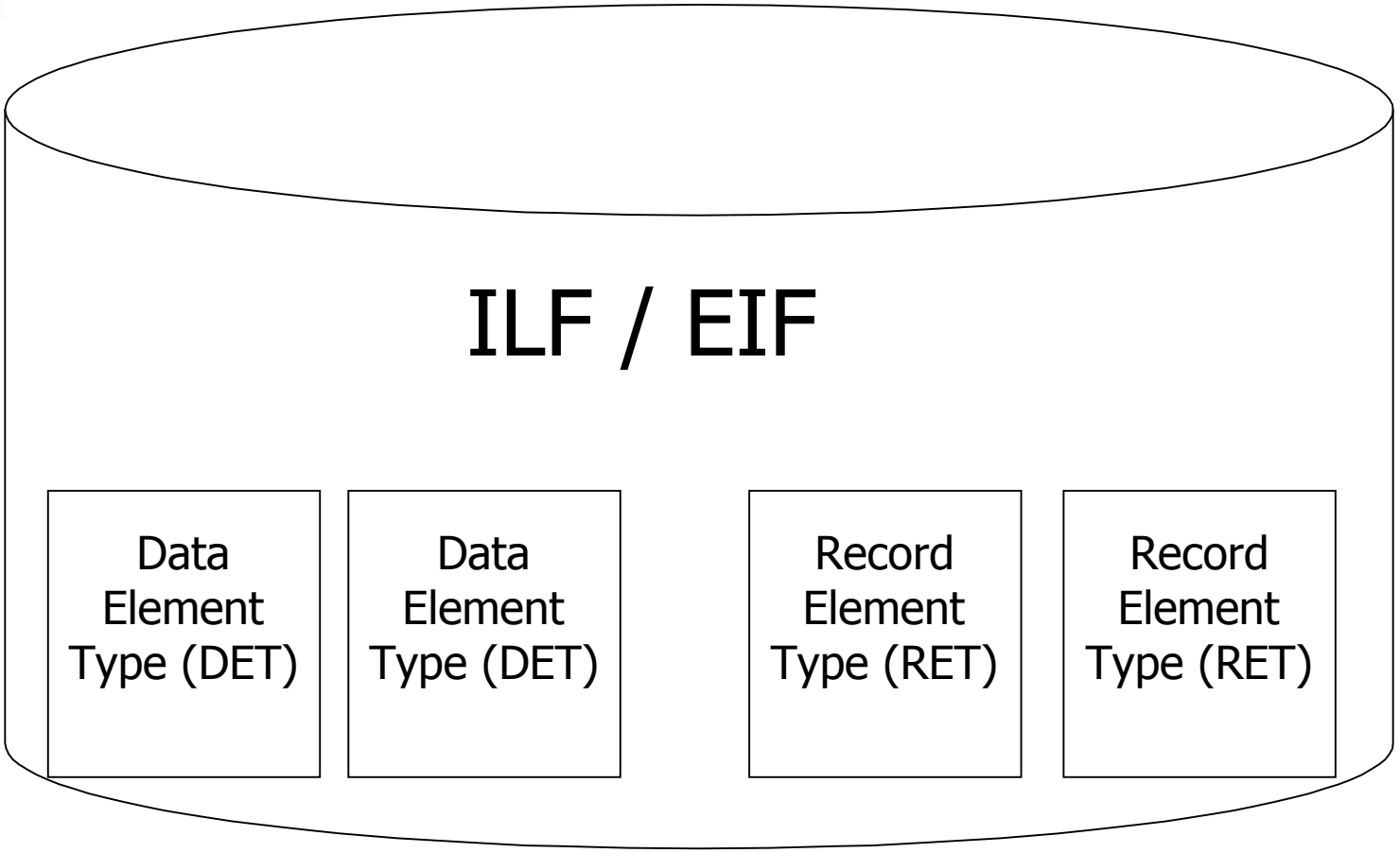
External Interface File

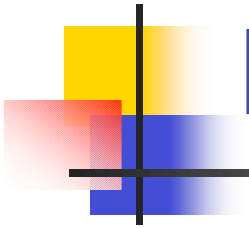
A EIF is a group of user identifiable, logically related information which is:

- Data or Control Information
- referenced by the application
- maintained by another application

- Primary Intent: Hold Data outside Application

Complexity and Contribution



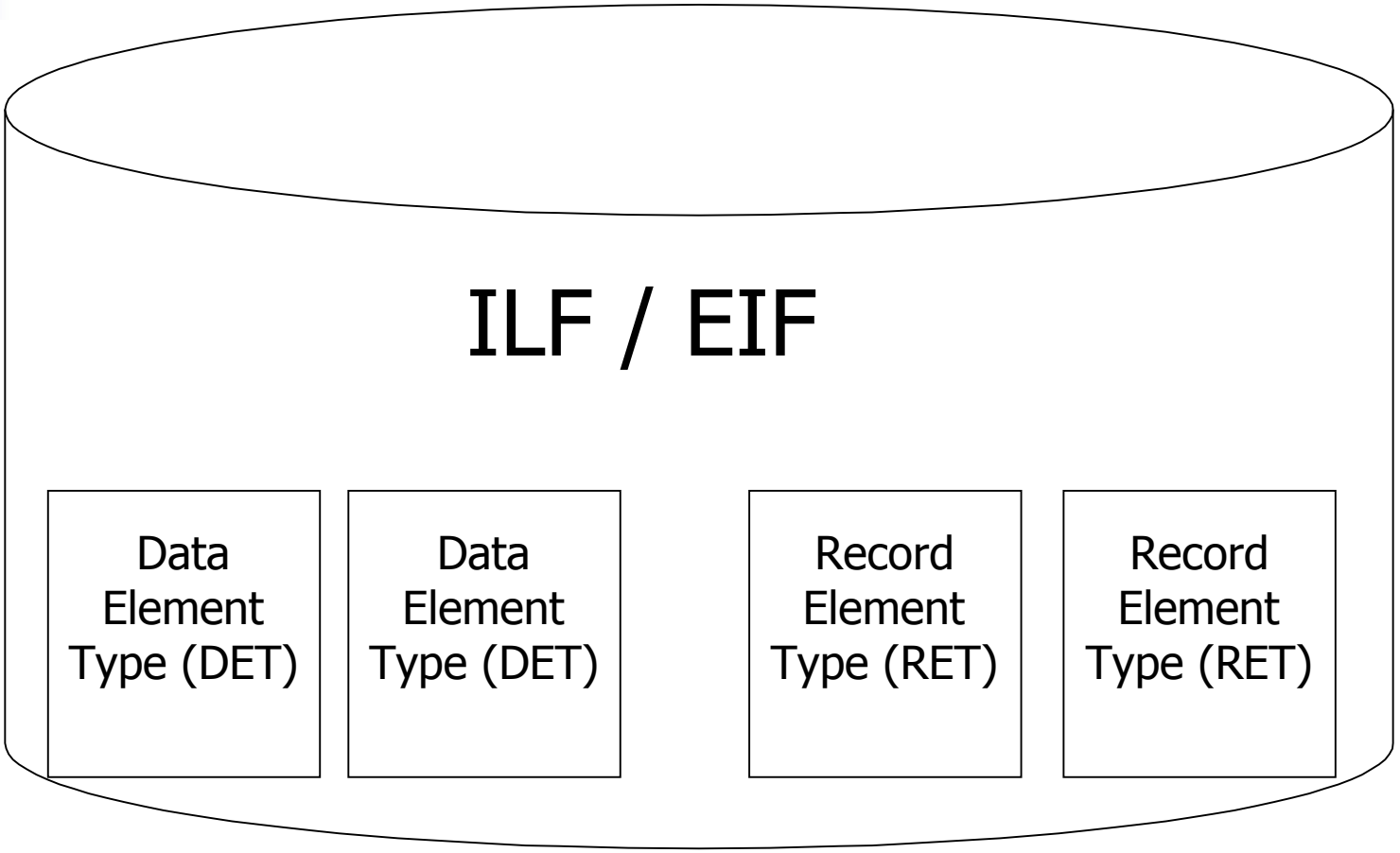


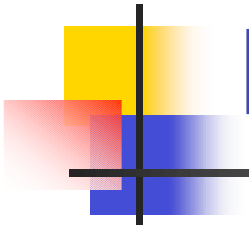
Data Element Type

A DET is a field of an EIF/ILF which is:

- user recognizable
- nonrecursive
- Maintained in or retrieved from the File through an elementary process

Complexity and Contribution



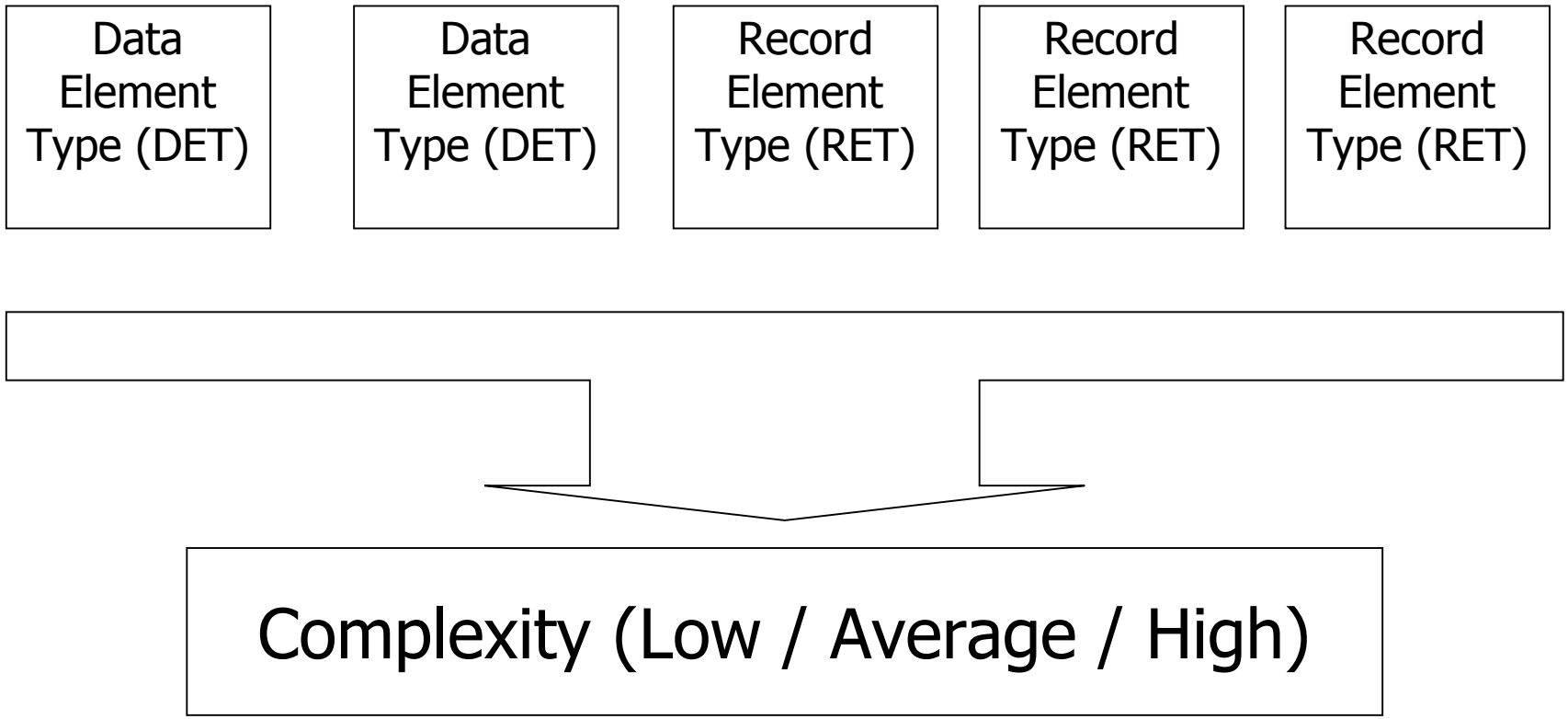


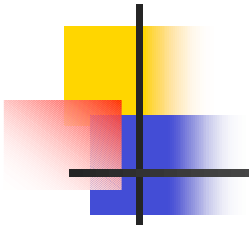
Record Element Type

A RET is a subgroup of data elements of an ILF/EIF which is:

- User recognizable
- Either optional or mandatory

Complexity

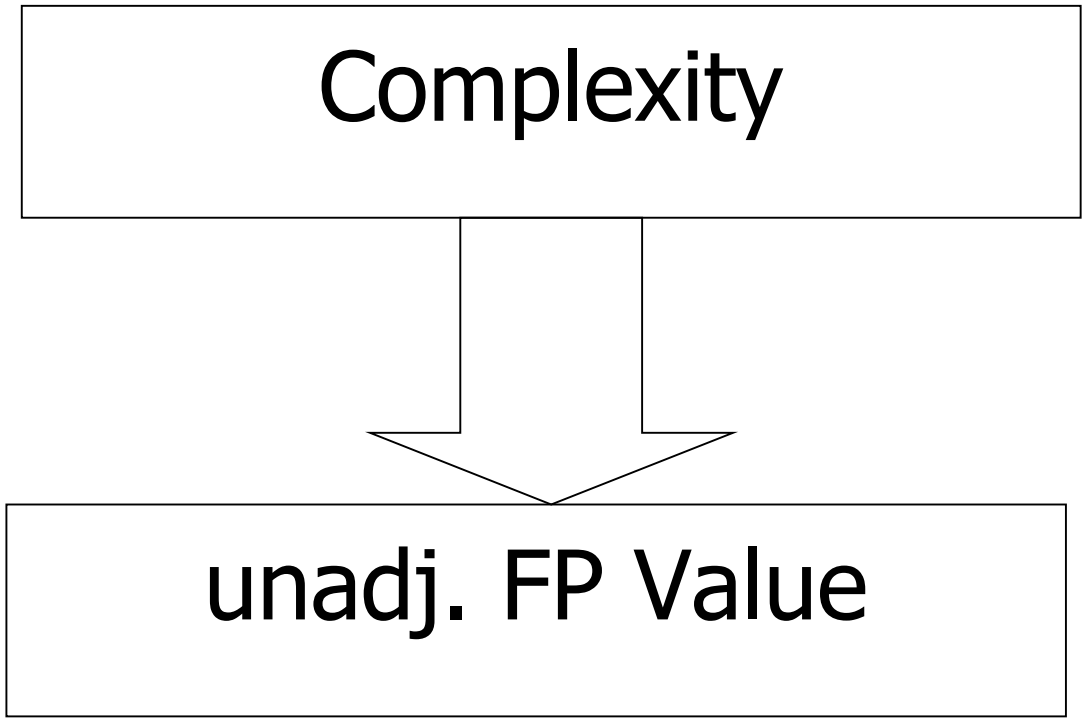

Data Element Type (DET)
Data Element Type (DET)
Record Element Type (RET)
Record Element Type (RET)
Record Element Type (RET)
Complexity (Low / Average / High)



Complexity

	1 – 19 DET	20 – 50 DET	51 + DET
1 RET	Low	Low	Average
2 – 5 RET	Low	Average	High
6 + RET	Average	High	High

Contribution



Contribution ILF / EIF

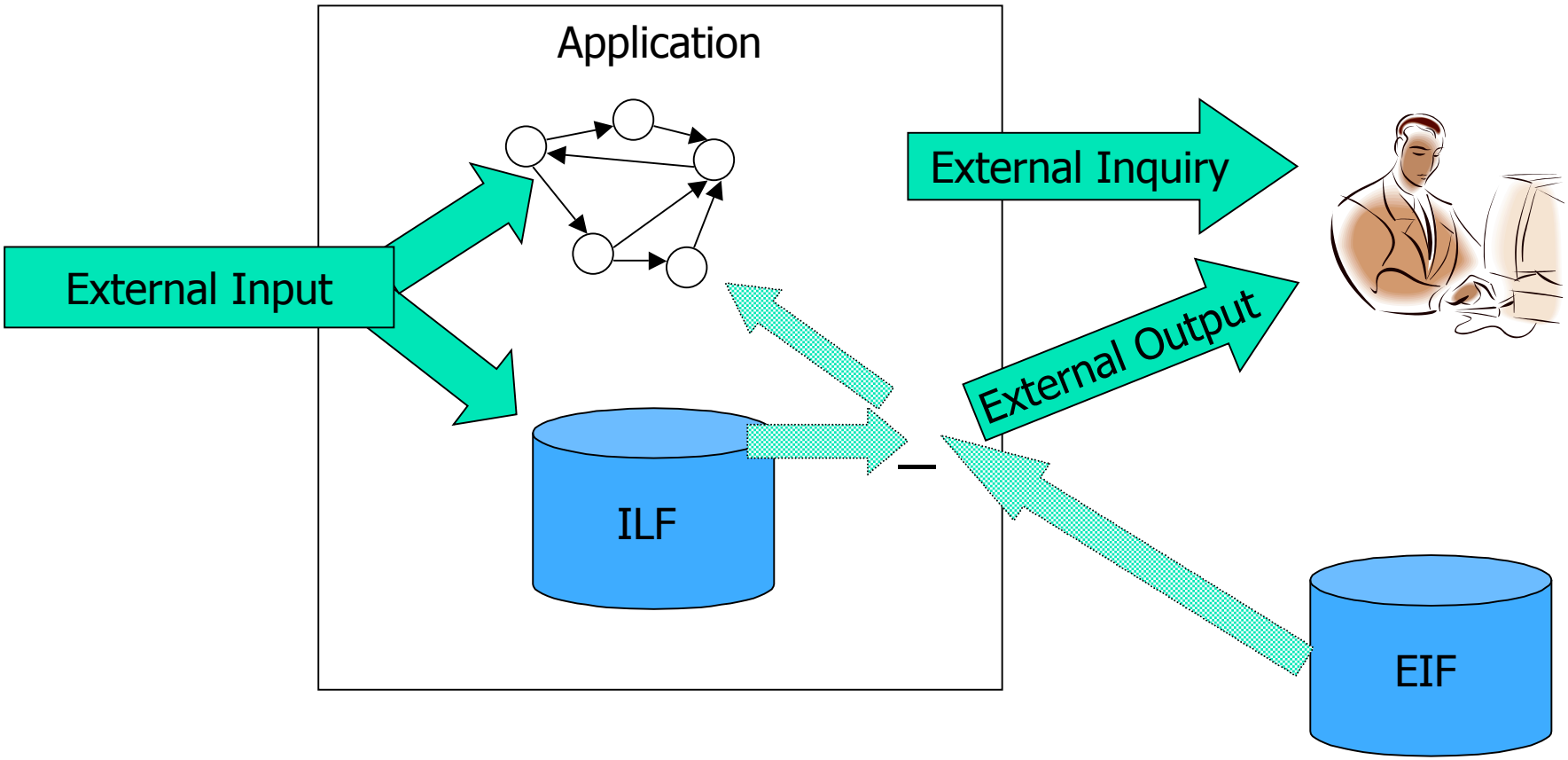
Internal Logical File	
Complexity Rating	Unadj. FPs
Low	7
Average	10
High	15

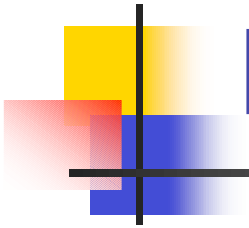
External Interface File	
Complexity Rating	Unadj. FPs
Low	5
Average	7
High	10

Function Type Totals

Function Type	Functional Complexity				Function Type Totals
	#	Complexity	X		
ILF	0	Low	7	0	
	0	Average	10	0	
	1	High	15	15	15
EIF	0	Low	5	0	
	2	Average	7	14	
	1	High	10	10	24

Transactional Functions

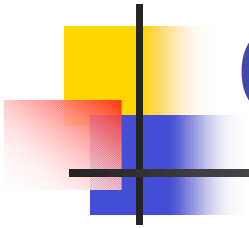




External Input Rules

- Receives data or control information from outside the boundary through an elementary process
- Information is either:
  - _ Altering the System State
  - _ Maintaining one or more ILF
- The different EIs are disjunctive in functionality and Datafiles

External Inquiry / External Output shared Rules



- Sends data or control information outside the boundary through an elementary process
- contains a unique processing logic

OR

- contains unique data elements

OR

- references ILFs or EIFs different from other EOs or EQs

External Output additional Rules

- Additional rules:
- Processing logic contains at least one mathematical formula

OR

- Processing logic creates derived data

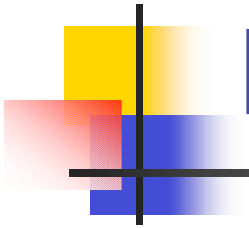
OR

- Processing logic maintains one or more ILFs

OR

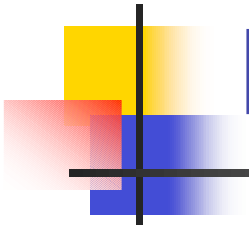
- Processing logic alters system state

External Inquiry additional Rules



- Processing Logic (PL) retrieves data or control information from an ILF or EIF
- PL does not contain mathematical formulas
- PL does not create derived data
- PL does not maintain an ILF
- PL does not alter system state

Complexity and Contribution: FTRs

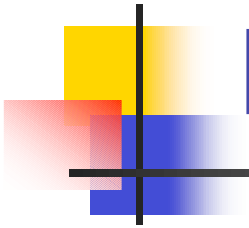


- File type referenced: A ILF read or maintained by the transactional function

OR

- A EIF read by the transactional function

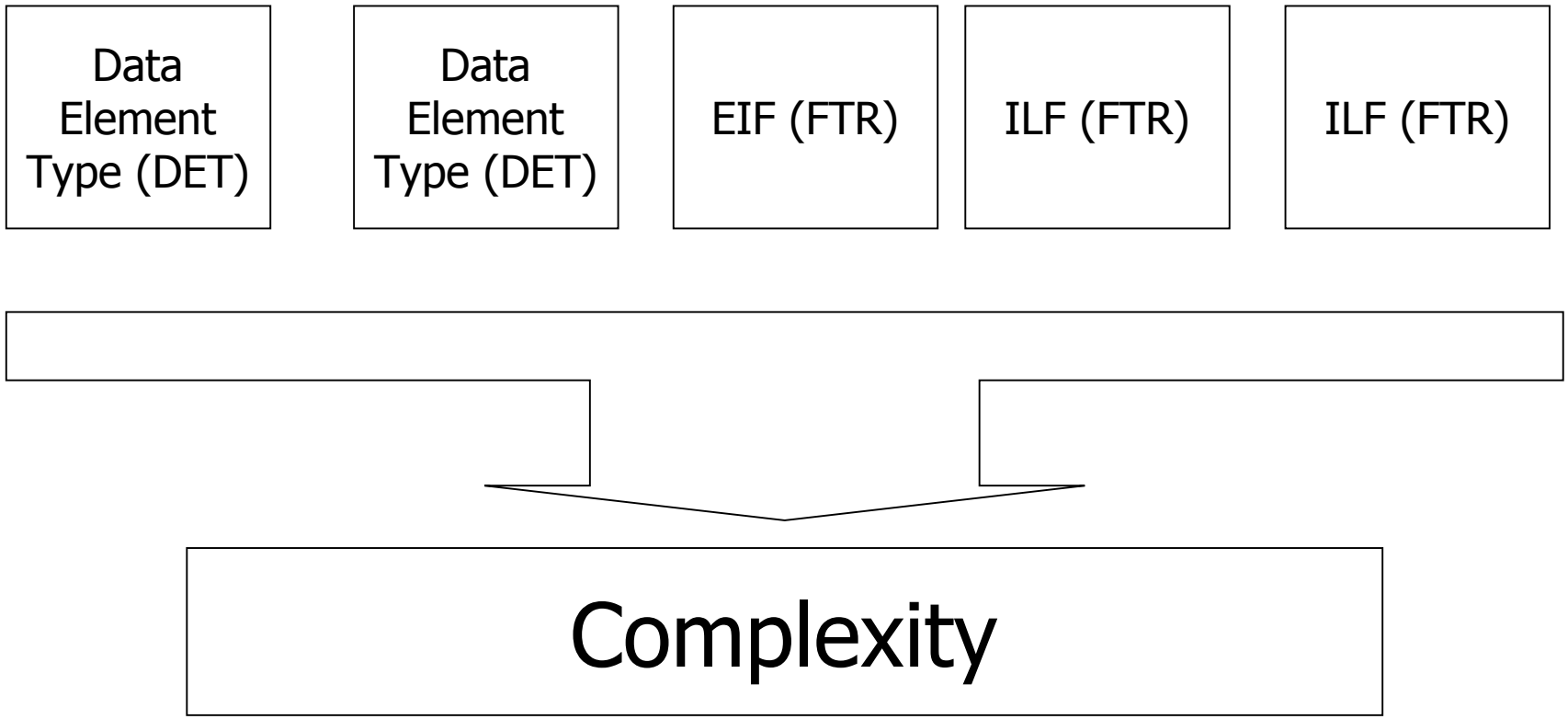
Complexity and Contribution: DETs



Data element type:

- Crossing the boundary
- necessary for the Process
- for sending a system response message out of the boundary
- for the ability to specify an action to be taken

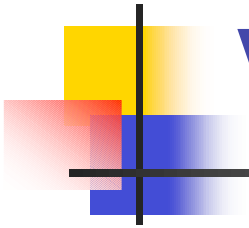
Complexity


Data Element Type (DET)
Data Element Type (DET)
EIF (FTR)
ILF (FTR)
ILF (FTR)
Complexity

Complexity: EIs

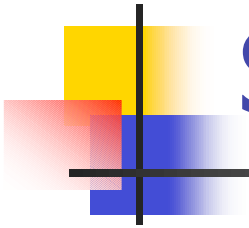
External Inputs			
	1 – 4 DET	5 – 15 DET	16 + DET
0 – 1 FTR	Low	Low	Avg.
2 FTR	Low	Avg.	High
3 + FTR	Avg.	High	High

External External Outputs / Inquiries			
	1 – 5 DET	6 – 19 DET	19 + DET
0 – 1 FTR	Low	Low	Avg.
2 – 3 FTR	Low	Avg.	High
4 + FTR	Avg.	High	High



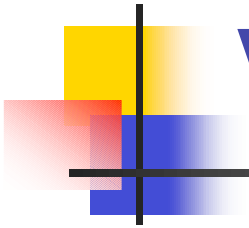
Value Adjustment Factor

- Used for rating the general functionality of the application
- Depends on characteristics and context
- 14 system characteristics (IFPUG), each between 0 and 5
- May adjust unadjusted FP value up to +/- 35%



System Characteristics

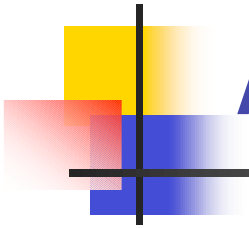
1. Data Communications
2. Distributed Data Processing
3. Performance
4. Heavily Used Configuration
5. Transaction Rate
6. Online Data Entry
7. End-User Efficiency
8. Online Update
9. Complex Processing
10. Reusability
11. Installation Ease
12. Operational Ease
13. Multiple Sites
14. Facilitate Change



Value adjustment Factor

- Sum up all 14 degrees of influence for the system characteristics → Total degree of influence (TDI)

- VAF = (TDI * 0.01) +0.65
- VAF = [0.65..1.35]



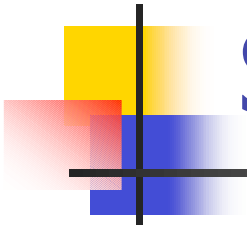
Adjusted FP Value

- FP = (UFP + CFP) * VAF

- UFP: Unadjusted FP value
- CFP: Conversion functionality FP value (unadjusted)
- VAF: Value adjustment Factor

Advantages and Disadvantages of the FP Method

Advantages	Disadvantages
FPs are independent of: ▪ Hardware ▪ Language ▪ Platform	No standard exists, but more than 35 different methods
FPs are understandable /comparable for non technical users	Only trained people can use the method accurately
Method can be applied early in the specification process, only basic requirements are needed	Relatively high effort and high cost
Very reliable for projects of medium and large size	Not useable for small projects
extendable	



Summary

The Function Point Method is an excellent tool for estimating effort for Software Projects. It has been proven to be very reliable for medium and large Projects. The result is independent of Hard- and Software and based on the user's view.

Alas there are way too many different methods for counting FPs and the results differ up to 50% or more. Solution: Standards (IFPUG or ISO)