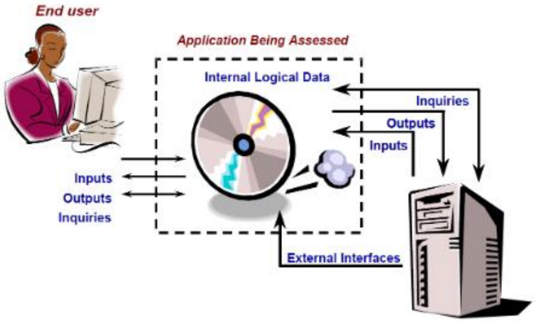
Function Points Calculation



Ahsan Riaz

Basis of cost estimation:

Historical data

Measure of work—SLOC/KLOC Source Lines of Code (SLOC) KLOC = The size of the code for the project in Kilo lines of code.

Complexity— even same SLOCs but complexity varies

Cost Estimation Tools and Techniques:

- Analogous Estimate
- Bottom up estimate
- **Parametric modeling**

Parametric modeling

- Uses project characteristics in mathematical model to estimate project costs
- E.g..$ 50 per SLOC
- Most reliable when historical information is accurate.
- COCOMO,COCOMOII are examples
- Other computerized tools available for cost estimation such as spreadsheets, PM software

Discussion:

Project Estimation Techniques

Project Size

Effort required to complete the Project

Duration

Cost

LOC & Function Points >> COCOMO

Discussion:

Function Points—no of SLOCs

Object Points—determine effort on person months

COCOMO model

LOC Line of Code

We can measure cost of a software project by using LOC, excluding comments etc, count the lines, but line of code varies from one programming language to another, it will be different for C++, Python, Java etc

So to overcome this problem we use Function Point Method as a Cost estimation Method

LOC productivity

- **Real-time embedded systems:**

  40-160 LOC/P-month

- **Systems programs:**

  150-400 LOC/P-month

- **Commercial applications:**

  200-800 LOC/P-month

- **Object-points productivity: PROD**

  measured 4 -50object points/person-month

  depends on tool support and developer capability

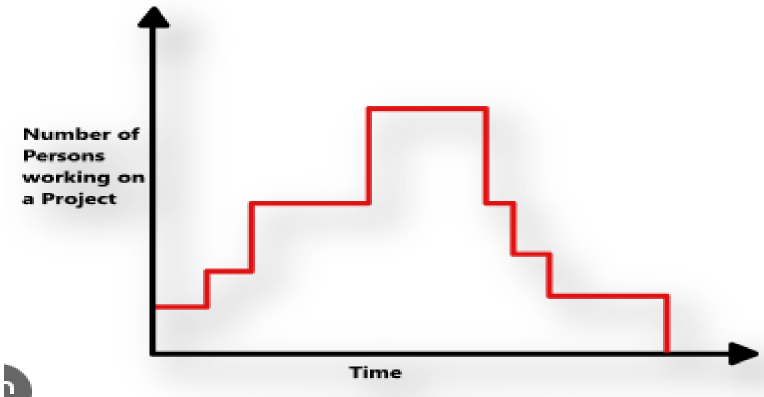


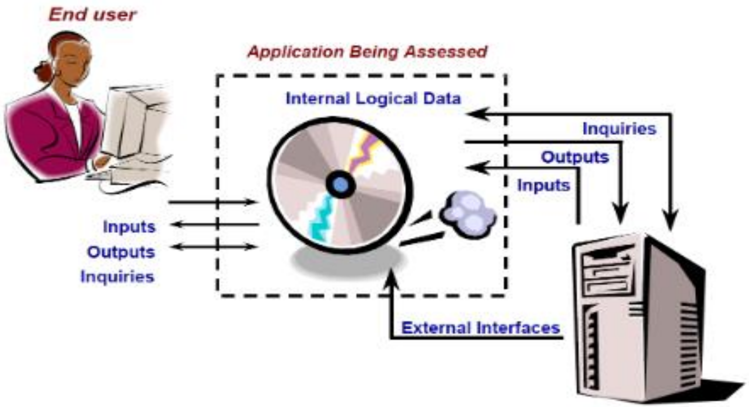
Fig. Person Month Curve

Function Points

- Quantify the functional size of programs independently of the programming languages
- Presented by A.J. Albrecht of IBM In the late1970‘s.

Function Point:

FP measures functionality from the User's point of view like what the user receives from the software and what the user requests from the software. It focuses on what functionality is being delivered.



Why use Function Points:

It’s really difficult to weigh the real values or worth of software. One of the best ways to estimate the software development cost and time is based on the previous software development experience.

To measure the standard worth of the software, as a unit of software worth, Function Point was developed. Functional Point was first defined by Allan Albrecht of IBM in 1977. It is used effectively as a means for measuring the functionality delivered by a system.

Using historical data, the FP metric can then be used to

1. Estimate the cost or effort required to design, code, and test the software.

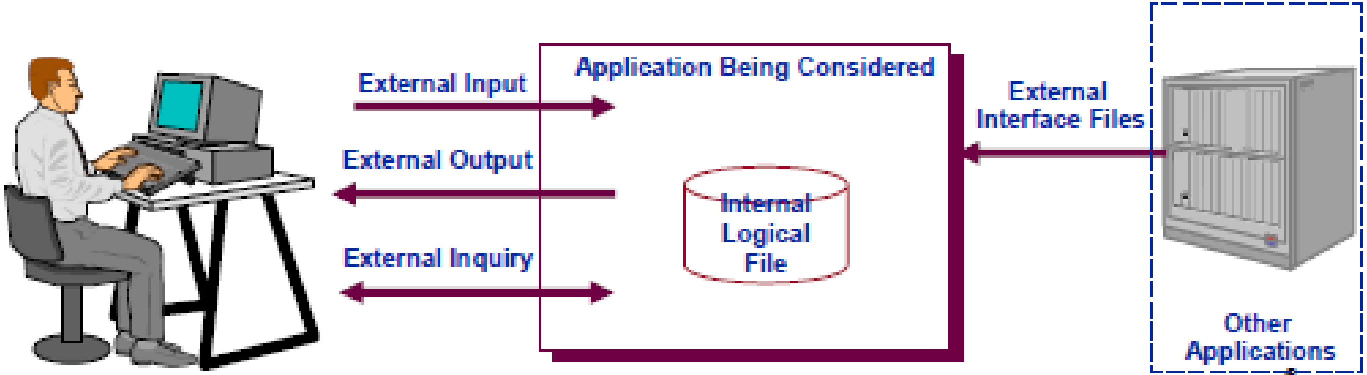
2. Predict the number of errors that will be encountered during testing

3. Forecast the number of components and/or the number of projected source lines in the implemented system.

FPs are derived using an empirical relationship based on countable measures of software’s information domain and qualitative assessments of software complexity.

The Five Components of Function Points

- Data Handling Functions
  - Internal Logical Files
  - External Interface Files
- Data Transactional functions
  - External Inputs
  - External Outputs
  - External Inquiries



The **five functional** units which is considered information domain as input to calculate the FP are.

1. **Internal Logic Files (ILF)** – The control info or logically related data that is present within the system.
2. **External Interface Files (EIF)** – The control data or other logical data i.e referenced by the system but present in another system.
3. **External Inputs (EI)** – Data / control info that comes from outside our system
4. **External Outputs (EO)** – data that goes out of the system after generation
5. **External Enquired (EQ)** – Combination of i/o – o/p resulting data retrieval

Formula

To compute FP, the following relationship is used:

FP = UFP X CAF

UFP is Unadjusted Functional Point

CAF is Complexity Adjustment Factor

UFP = Sum of all the Complexities of all the EI's, EO's EQ's, ILF's and EIF's

CAF = 0.65 + (0.01 x summation Fi)

0-5 scale

0 – No Influences or no important or no applicable

1 – Incidental

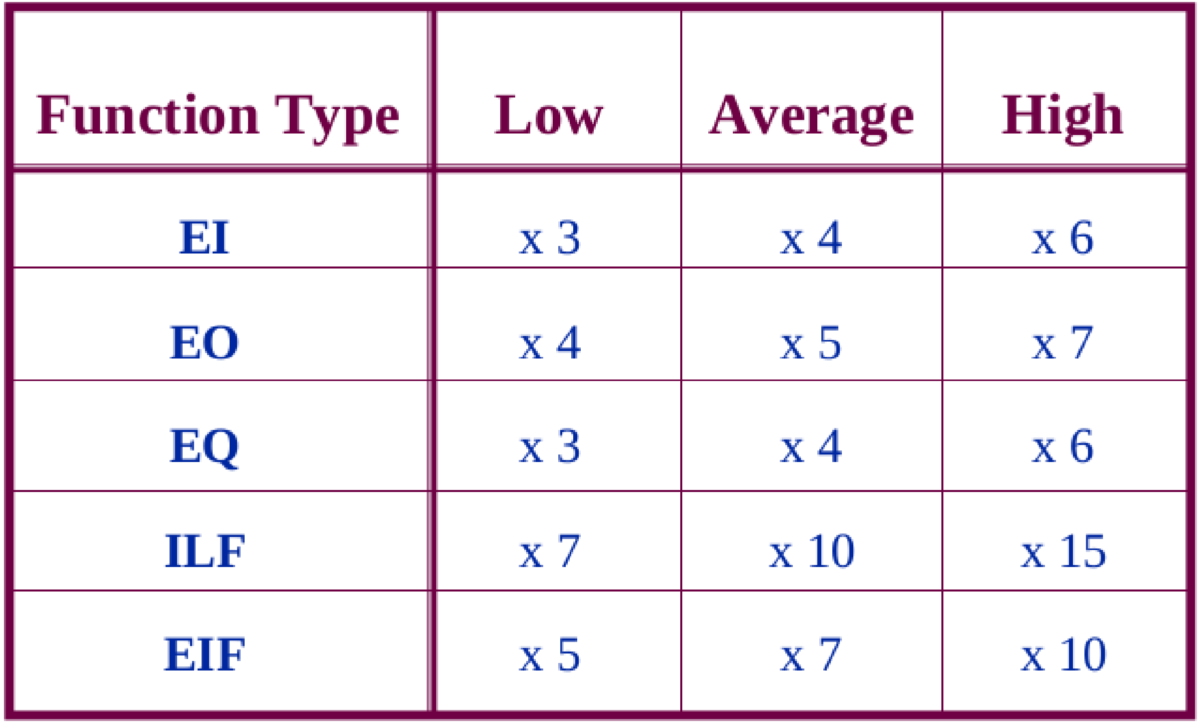
2 – Moderate

3 – Average

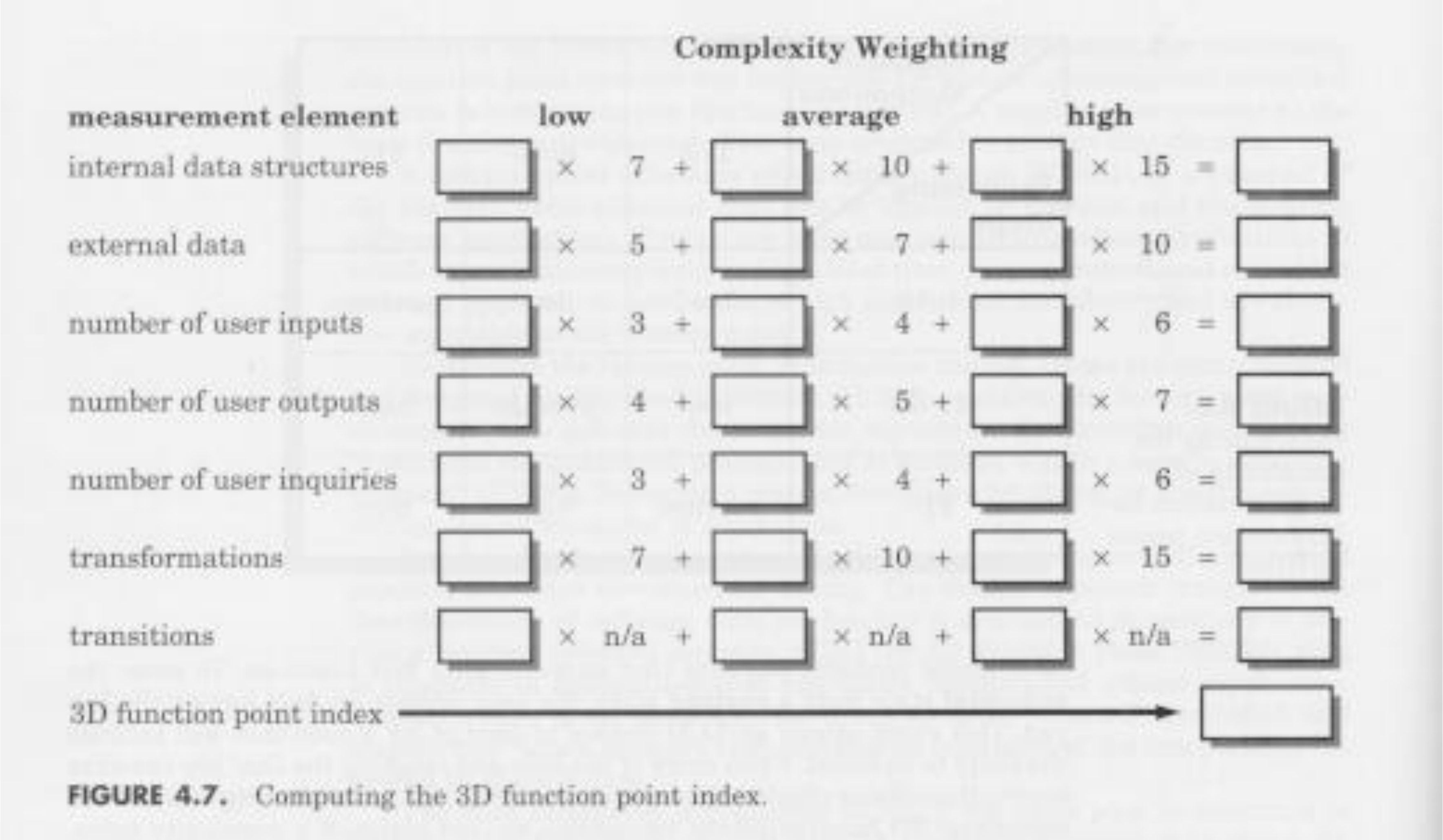
4 – Significant

5 – Essential

Functional Size
(Unadjusted Function Size)



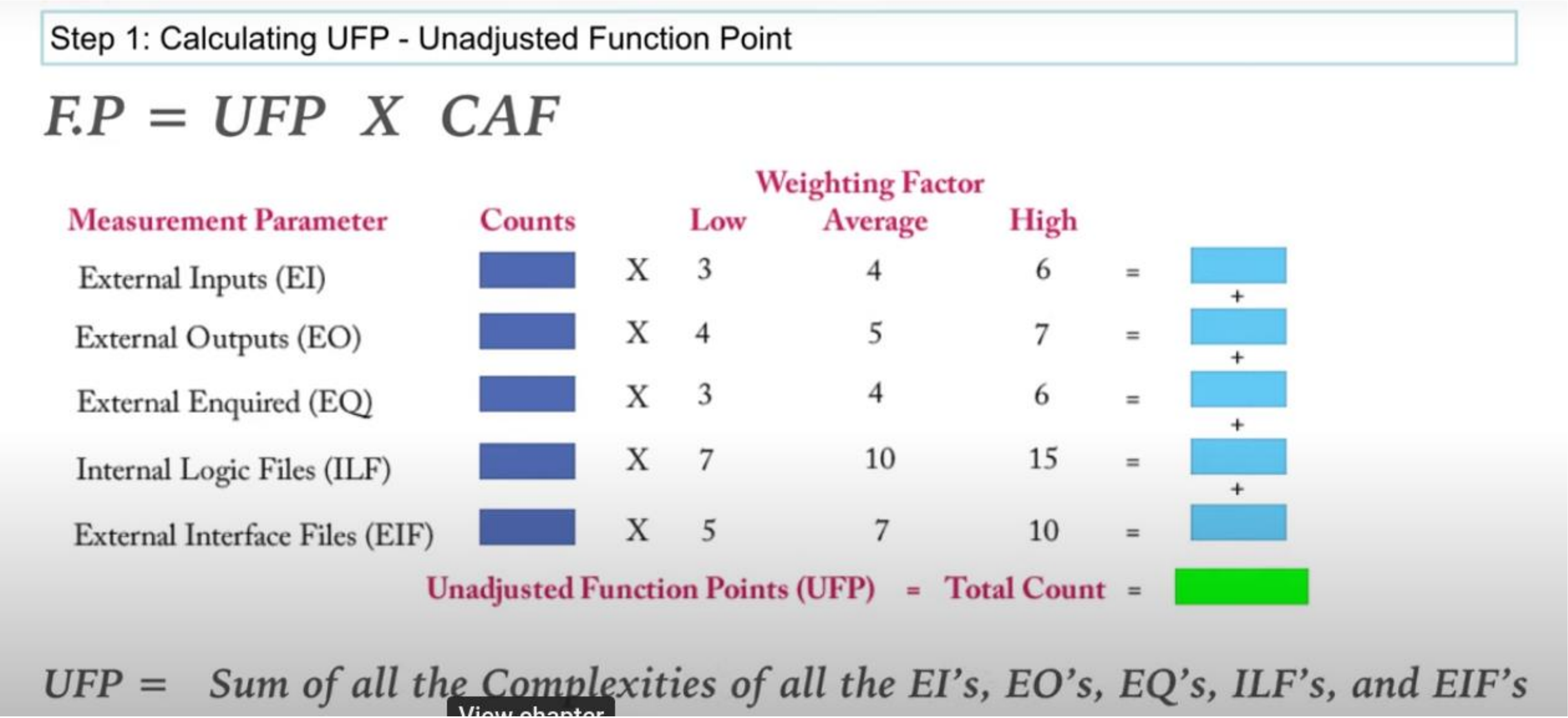
Function Type	Low	Average	High
EI	x 3	x 4	x 6
EO	x 4	x 5	x 7
EQ	x 3	x 4	x 6
ILF	x 7	x 10	x 15
EIF	x 5	x 7	x 10



	Complexity Weighting			
measurement element	**low**	**average**	**high**	
internal data structures	□ × 7 +	□ × 10 +	□ × 15 =	□
external data	□ × 5 +	□ × 7 +	□ × 10 =	□
number of user inputs	□ × 3 +	□ × 4 +	□ × 6 =	□
number of user outputs	□ × 4 +	□ × 5 +	□ × 7 =	□
number of user inquiries	□ × 3 +	□ × 4 +	□ × 6 =	□
transformations	□ × 7 +	□ × 10 +	□ × 15 =	□
transitions	□ × n/a +	□ × n/a +	□ × n/a =	□
3D function point index →				□

FIGURE 4.7. Computing the 3D function point index.

Now UFP



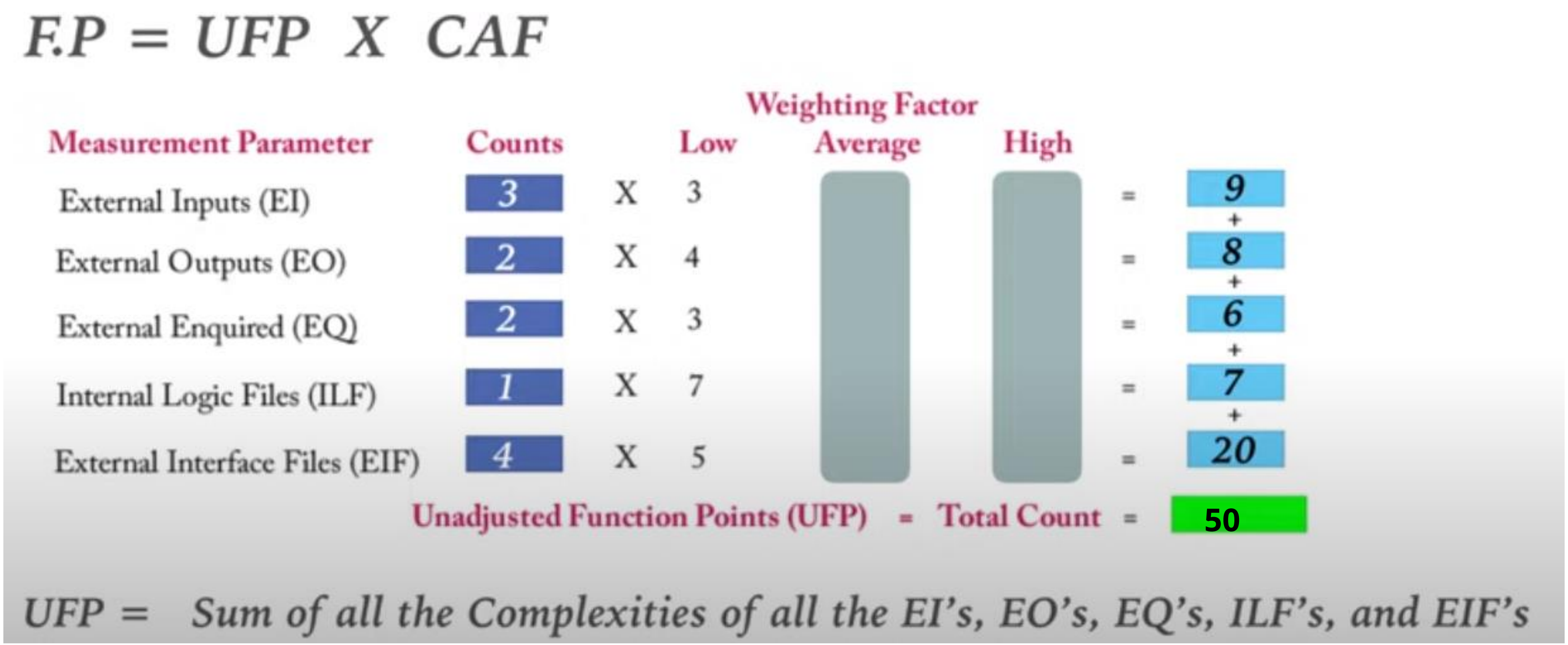
Step 1: Calculating UFP - Unadjusted Function Point

$$F.P = UFP \times CAF$$

			Weighting Factor				
Measurement Parameter	Counts		Low	Average	High		
External Inputs (EI)	🟦	X	3	4	6	=	🟦 +
External Outputs (EO)	🟦	X	4	5	7	=	🟦 +
External Enquired (EQ)	🟦	X	3	4	6	=	🟦 +
Internal Logic Files (ILF)	🟦	X	7	10	15	=	🟦 +
External Interface Files (EIF)	🟦	X	5	7	10	=	🟦

Unadjusted Function Points (UFP) = Total Count = 🟩

UFP = Sum of all the Complexities of all the EI's, EO's, EQ's, ILF's, and EIF's

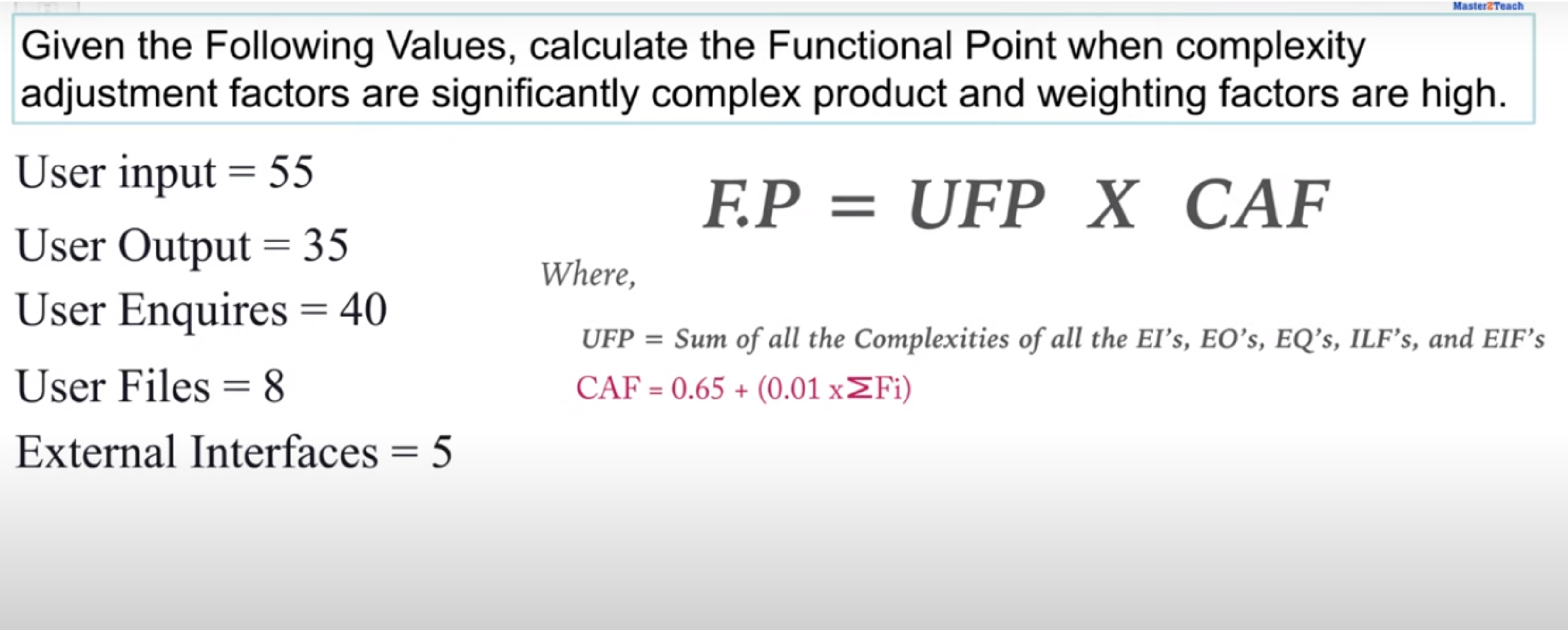


$F.P = UFP \ X \ CAF$

			Weighting Factor				
Measurement Parameter	Counts		Low	Average	High		
External Inputs (EI)	3	X	3			=	9
							+
External Outputs (EO)	2	X	4			=	8
							+
External Enquired (EQ)	2	X	3			=	6
							+
Internal Logic Files (ILF)	1	X	7			=	7
							+
External Interface Files (EIF)	4	X	5			=	20

Unadjusted Function Points (UFP) = Total Count = 50

UFP = *Sum of all the Complexities of all the EI's, EO's, EQ's, ILF's, and EIF's*



Given the Following Values, calculate the Functional Point when complexity adjustment factors are significantly complex product and weighting factors are high.

User input = 55

User Output = 35

User Enquires = 40

User Files = 8

External Interfaces = 5

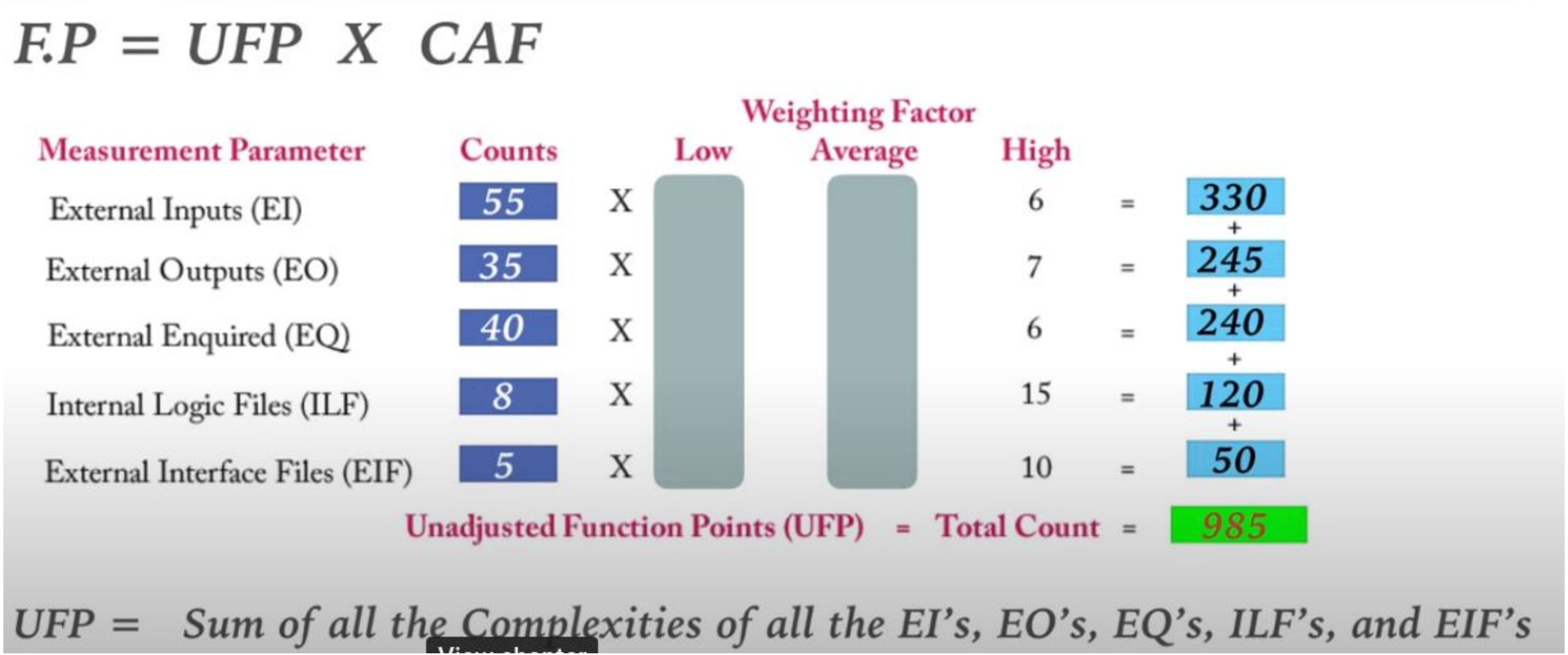
$$F.P = UFP \ X \ CAF$$

Where,

UFP = Sum of all the Complexities of all the EI's, EO's, EQ's, ILF's, and EIF's

$CAF = 0.65 + (0.01 \ x \Sigma Fi)$

Step 1: Calculating UFP



$$F.P = UFP \ X \ CAF$$

Measurement Parameter	Counts		Weighting Factor				
			Low	Average	High		
External Inputs (EI)	55	X			6	=	330 +
External Outputs (EO)	35	X			7	=	245 +
External Enquired (EQ)	40	X			6	=	240 +
Internal Logic Files (ILF)	8	X			15	=	120 +
External Interface Files (EIF)	5	X			10	=	50

Unadjusted Function Points (UFP) = Total Count = 985

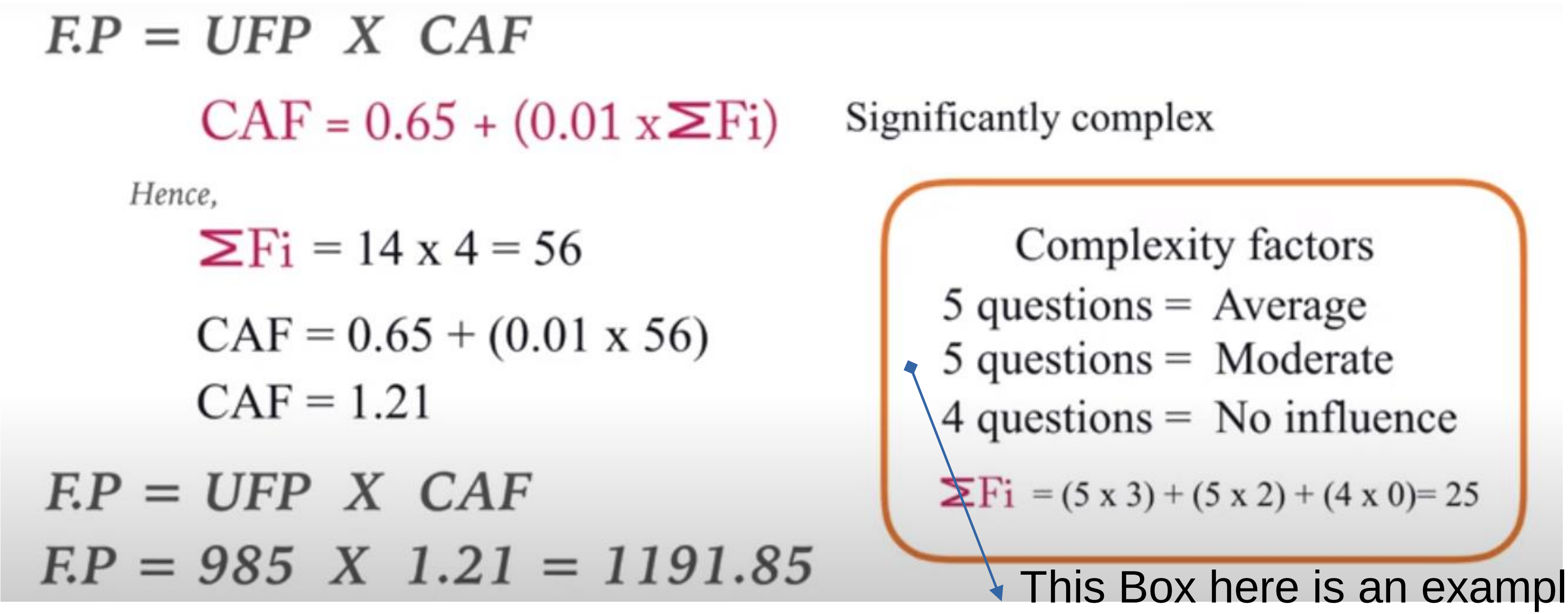
UFP = Sum of all the Complexities of all the EI's, EO's, EQ's, ILF's, and EIF's

CAF

CAF = 0.65 + (0.01 x submission Fi)

Where Fi is value adjustment factors based on responses to the 14 questions

Complexity Adjustment Factor is calculated using 14 aspects of processing complexity and these 14 questions answered on a scale of 0 – 5



$$F.P = UFP \ X \ CAF$$

$$CAF = 0.65 + (0.01 \ x \sum Fi)$$

Significantly complex

Hence,

$$\sum Fi = 14 \ x \ 4 = 56$$

$$CAF = 0.65 + (0.01 \ x \ 56)$$

$$CAF = 1.21$$

$$F.P = UFP \ X \ CAF$$

$$F.P = 985 \ X \ 1.21 = 1191.85$$

Complexity factors

5 questions = Average

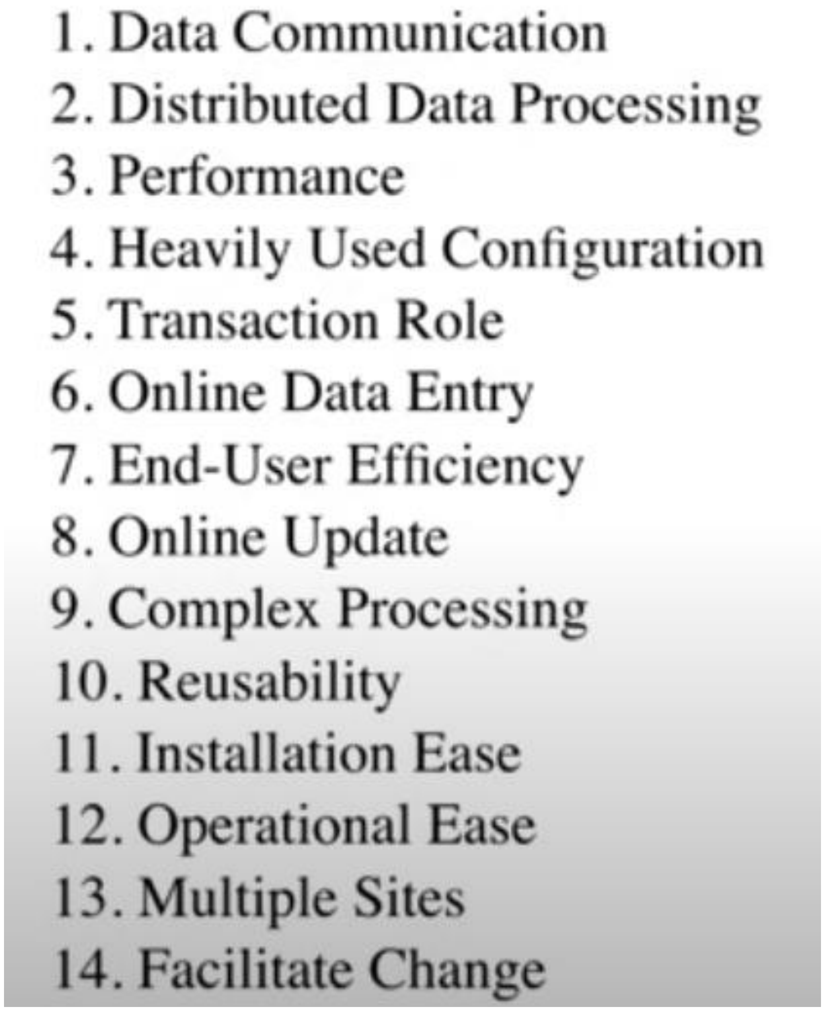
5 questions = Moderate

4 questions = No influence

$$\sum Fi = (5 \ x \ 3) + (5 \ x \ 2) + (4 \ x \ 0) = 25$$

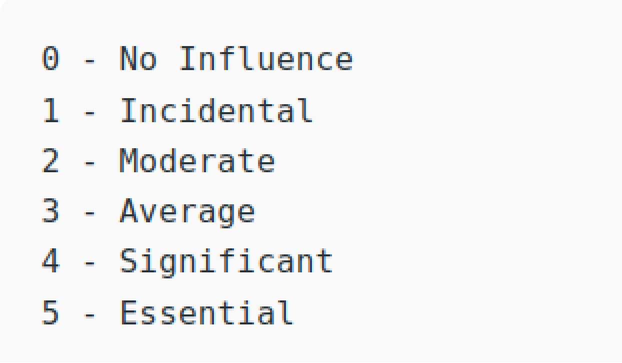
Here in this question all complexity factors are significant (Complex) or in simple words Significant means value is 4 So all 14 questions will be multiplied by 4

This Box here is an example that different Complexity Factors Have different values So how can you find Summation Fi value

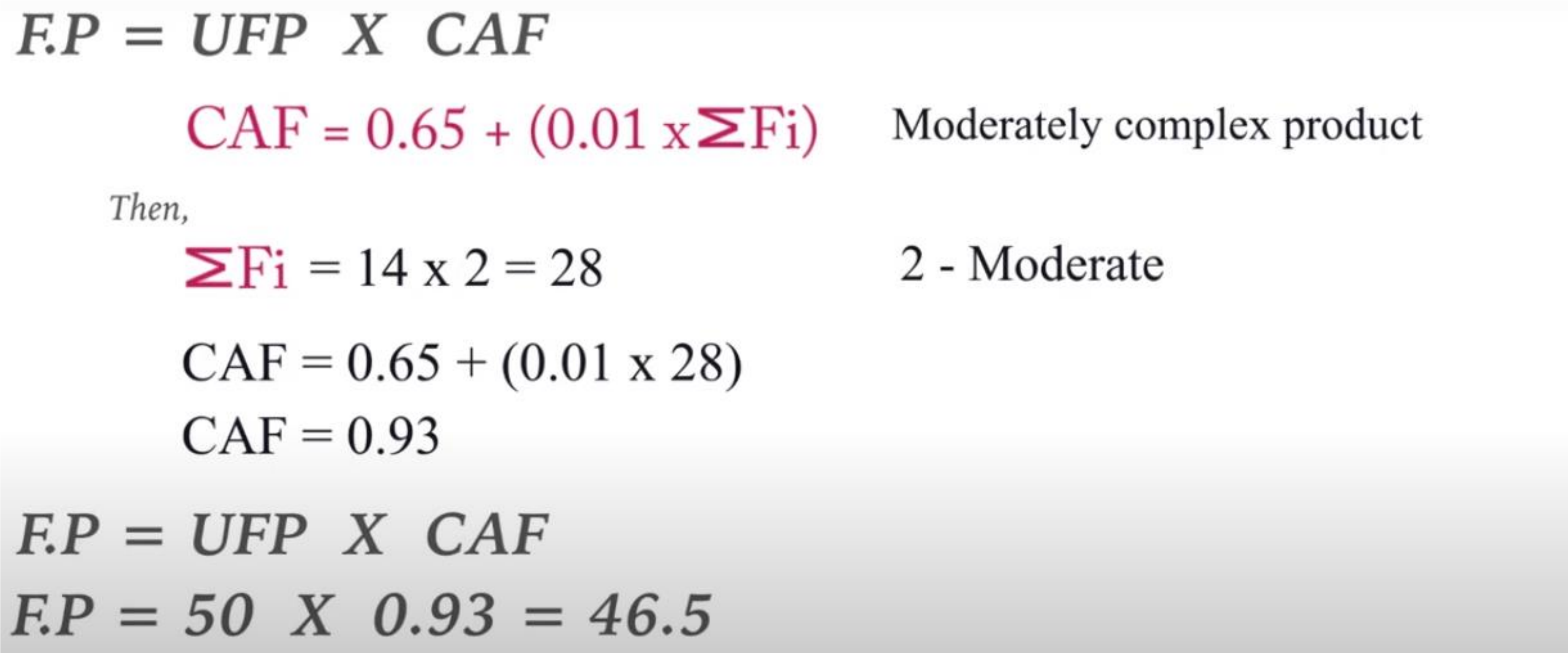


1. Data Communication
2. Distributed Data Processing
3. Performance
4. Heavily Used Configuration
5. Transaction Role
6. Online Data Entry
7. End-User Efficiency
8. Online Update
9. Complex Processing
10. Reusability
11. Installation Ease
12. Operational Ease
13. Multiple Sites
14. Facilitate Change

0-5 SCALE



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



$F.P = UFP \ X \ CAF$

$CAF = 0.65 + (0.01 \text{ x} \sum Fi)$ Moderately complex product

Then,

$\sum Fi = 14 \text{ x } 2 = 28$ 2 - Moderate

$CAF = 0.65 + (0.01 \text{ x } 28)$

$CAF = 0.93$

$F.P = UFP \ X \ CAF$

$F.P = 50 \ X \ 0.93 = 46.5$