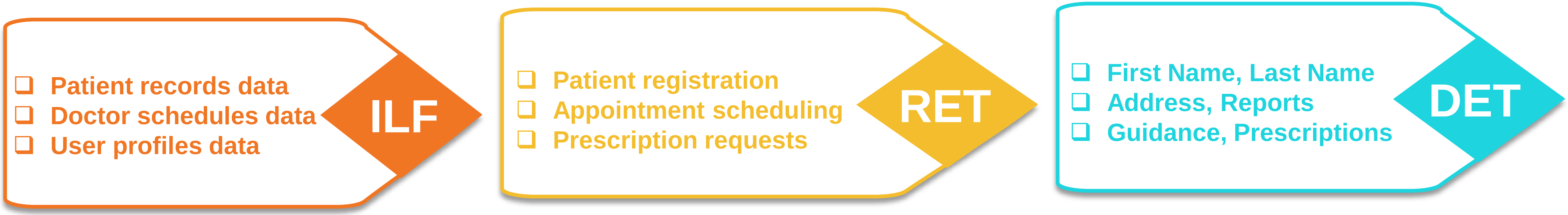
More about ILF, RET, and DET with out Project.

https://alvinalexander.com/FunctionPoints/node14.shtml
https://alvinalexander.com/FunctionPoints/node26.shtml

Internal Logical Files (ILFs) are data or information that the application uses and manages internally. These data or information can be stored in files, databases, or data structures used by the application. Examples of ILFs for a website like E-Doctor could be the patient's medical records, doctor's schedule, or user profiles.

External Inputs (RETS) are data or information that the application receives from external sources such as users, other systems, or devices. Examples of RETs for a website like E-Doctor could be a new patient registration, appointment scheduling, or prescription requests.

External Inquiries (DETs) are data or information that the application provides in response to user or system queries. Examples of DETs for a website like E-Doctor could be First name, Last Name, Address, Reports, Guidance and Prescriptions.



ILF
- □ Patient records data
- □ Doctor schedules data
- □ User profiles data

RET
- □ Patient registration
- □ Appointment scheduling
- □ Prescription requests

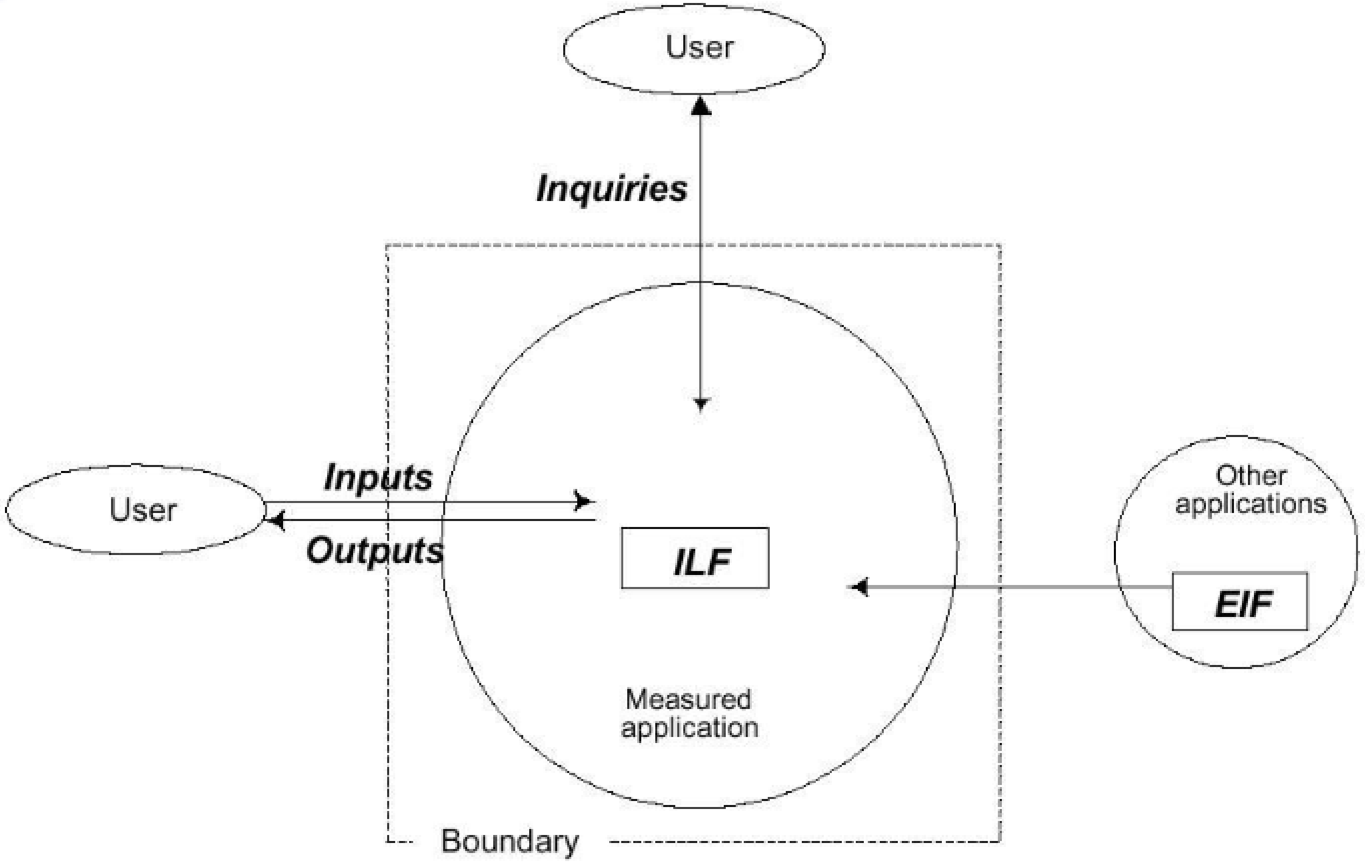
DET
- □ First Name, Last Name
- □ Address, Reports
- □ Guidance, Prescriptions

Transaction Function Points

- External Inputs
  - Information that comes from outside the application to inside the application boundary.
- External outputs
  - Information that crosses the boundary from inside to outside the application boundary that contains derived information or updates an internal logical file.
- External Inquiries
  - Information that crosses the boundary from inside to outside the application boundary that does not contain derived information or does not update an internal logical file.

Data Function Points:

- Internal Logical Files
  - A user identifiable group of logically related data that resides entirely within the applications boundary and is maintained through external inputs.
- External Interface Files
  - *A* user identifiable group of logically related data that is used for reference purposes only.


User
Inquiries
Inputs
User
Outputs
ILF
Other applications
EIF
Measured application
Boundary

Internal Logical Files
External Interface Files

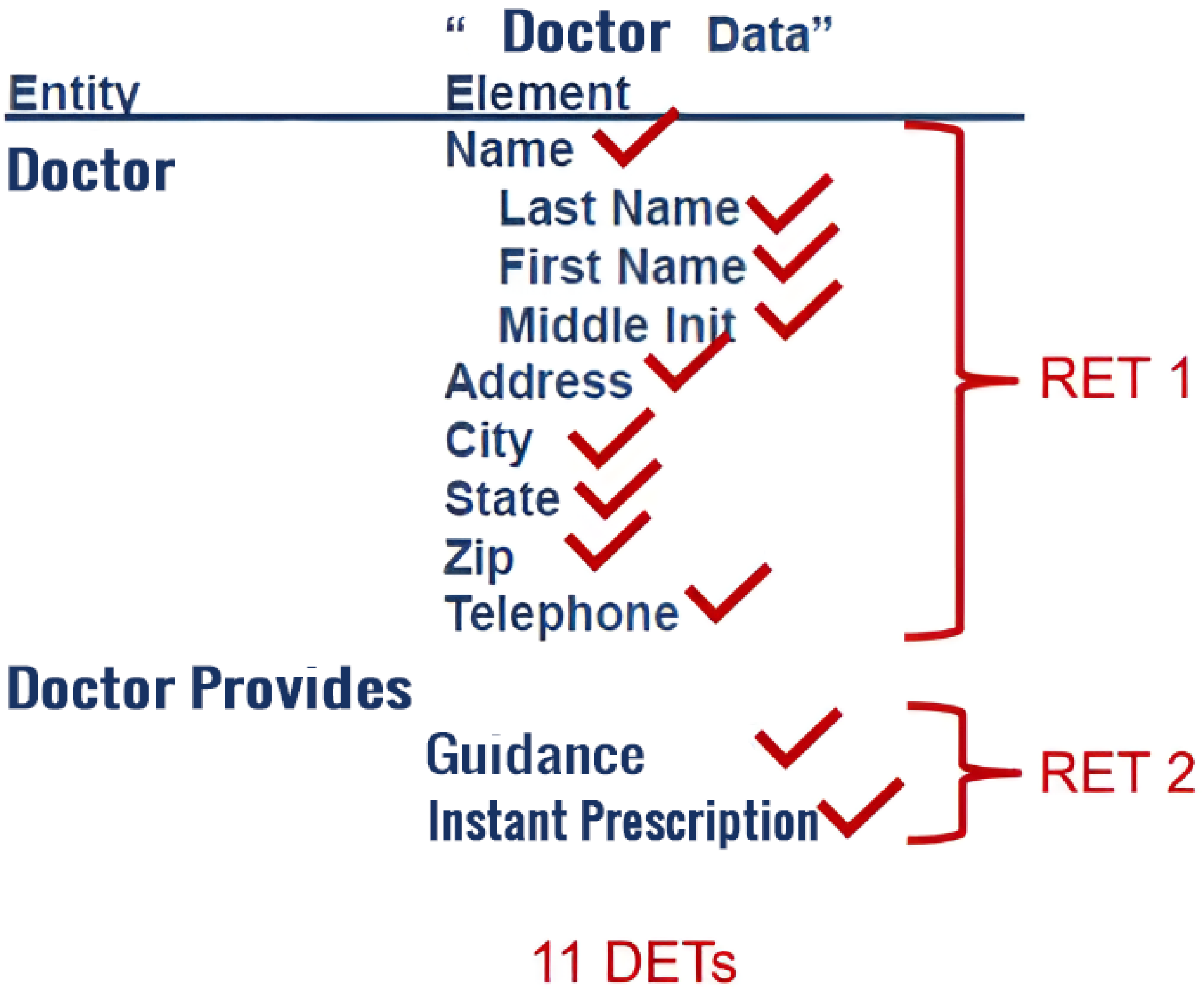
		DETs		
		1 - 19	20 - 50	51+
RETs	1	Low	Low	Average
	2 - 5	Low	Average	High
	6+	Average	High	High

- **Record Element Types (RETs)**

  Logical subgroups of data

- **Data Element Types (DETs)**

  Unique data fields



" Doctor Data"

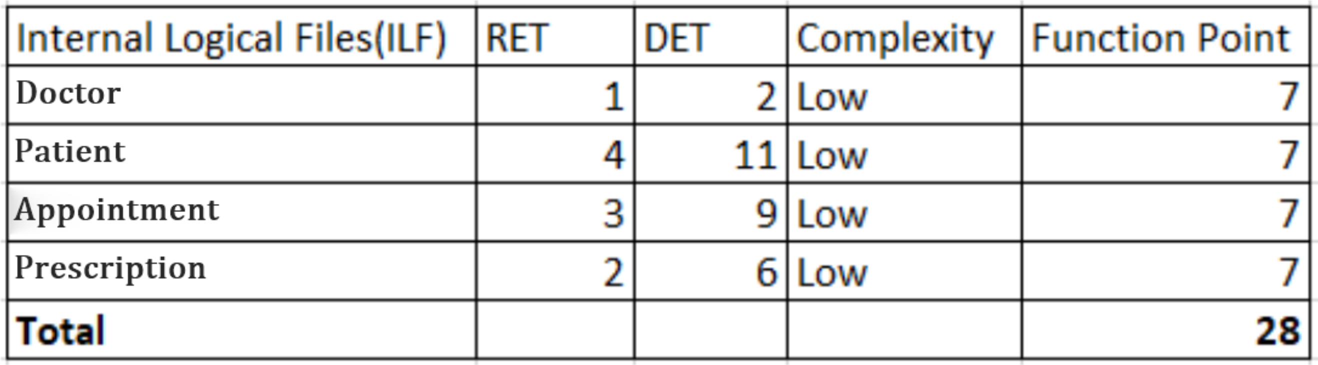
Entity	Element
Doctor	Name ✓
	Last Name ✓
	First Name ✓
	Middle Init ✓
	Address ✓
	City ✓
	State ✓
	Zip ✓
	Telephone ✓
Doctor Provides	
	Guidance ✓
	Instant Prescription ✓

11 DETs

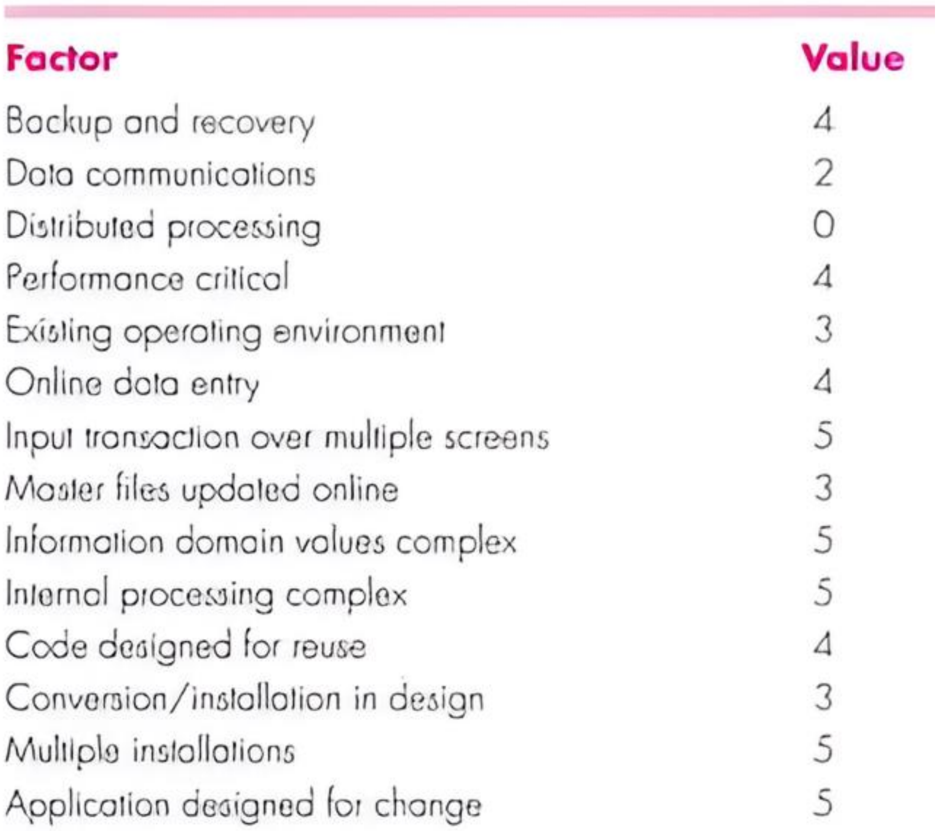
Measurement Parameter	Count	Simple	Average	Complex		Total
No of user inputs	7	3	4	6		28
No of user outputs	3	4	5	7		15
No of user inquiries	2	3	4	6		12
No of files	3	7	10	15		30
No of external interfaces	1	5	7	10		7
Count total (CT)						92

RETS	Data Element Types (DETs)		
	1-19	20-50	51+
1	L	L	A
2 to 5	L	A	H
6 or more	A	H	H

Complexity	Points
Low	7
Average	10
High	15

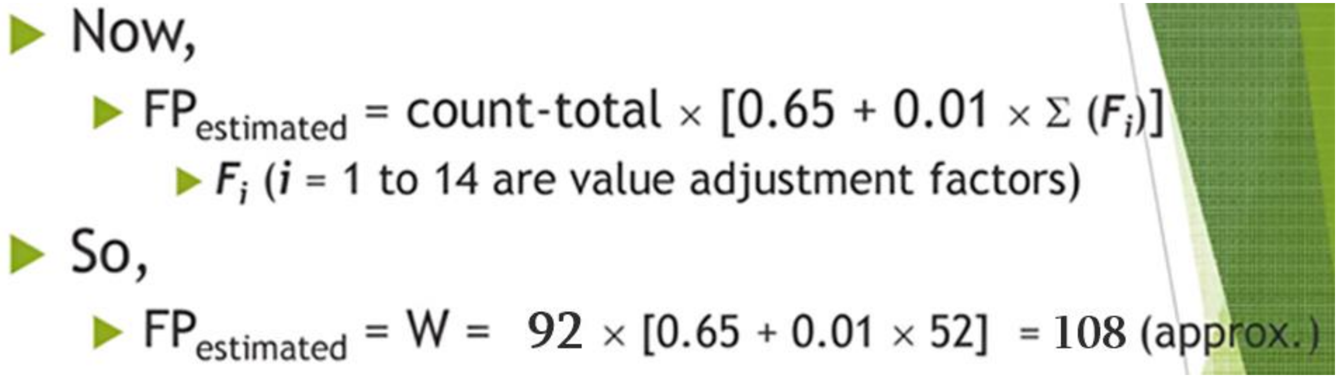


Internal Logical Files(ILF)	RET	DET	Complexity	Function Point
Doctor	1	2	Low	7
Patient	4	11	Low	7
Appointment	3	9	Low	7
Prescription	2	6	Low	7
Total				**28**



Factor	Value
Backup and recovery	4
Data communications	2
Distributed processing	0
Performance critical	4
Existing operating environment	3
Online data entry	4
Input transaction over multiple screens	5
Master files updated online	3
Information domain values complex	5
Internal processing complex	5
Code designed for reuse	4
Conversion/installation in design	3
Multiple installations	5
Application designed for change	5

Total =	**52**



- Now,
  - $FP_{estimated}$ = count-total × [0.65 + 0.01 × Σ (F_i)]
    - F_i (i = 1 to 14 are value adjustment factors)
- So,
  - $FP_{estimated}$ = W = 92 × [0.65 + 0.01 × 52] = 108 (approx.)

COCOMO MODEL

The Constructive Cost Model (COCOMO)

COCOMO stands for "Constructive Cost Model".
It is one of the very famous model which is used to estimate the cost of the project.

Using this model we can estimate the time(month) and no. of people (members) needed to develop a project.

COCOMO has three different models that reflect the complexity:

1. Basic Model
2. Intermediate Model
3. Detailed Model

The Development Modes: Project Characteristics

Organic Mode

1. Relatively small, simple software project

2. Small team with good application experience work to a set less than rigid requirement

3. Relatively small and requires little innovation

Semi Detached Mode

Intermediate (in size and complexity) software project in which teams with mixed experience levels must meet a mix of rigid requirement levels must meet a mix of rigid and less than rigid requirement.

Embedded Mode

Software project that must be developed within a set of tight hardware, software and operational constraint

BASIC COCOMO MODEL

It estimates the software **roughly** and **quickly**. It is mostly useful for **small** – **medium** sized software.

ORGANIC

SEMI-DETACHED

EMBEDED

Effort = $a\,(KLOC)^{b}$ *Person-Month*

Development Time = $c\,(Effort)^{d}$ *Months*

Average Staff Size = *Effort / Development Time* *Persons*

Productivity = *KLOC / Effort* *KLOC / Person - Month*

Type	*a*	*b*	*c*	*d*
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embeded	3.6	1.20	2.5	0.32

1. Organic

KLOC = 400

a = 2.4 b = 1.05 c = 2.5 d = 0.38

Effort

$$= a\ (KLOC)^{b} \quad \textit{Person-Month}$$

$$= 2.4\ (400)^{1.05} \quad \textit{Person-Month}$$

$$\approx 1295 \quad \textit{Person-Month}$$

Development Time

$$= c\ (Effort)^{d} \quad \textit{Months}$$

$$= 2.5\ (1295)^{0.38} \quad \textit{Months}$$

$$\approx 38 \quad \textit{Months}$$

2. Semi-detacheed

KLOC = 400

a = 3 b = 1.12 c = 2.5 d = 0.35

Effort

$$\begin{aligned} &= a\,(KLOC)^{b} && \textit{Person-Month} \\ &= 3\,(400)^{1.12} && \textit{Person-Month} \\ &\approx 2462 && \textit{Person-Month} \end{aligned}$$

Development Time

$$\begin{aligned} &= c\,(Effort)^{d} && \textit{Months} \\ &= 2.5\,(2462)^{0.35} && \textit{Months} \\ &\approx 38.4 && \textit{Months} \end{aligned}$$

3. Embeded

KLOC = 400

a = 3.6 b = 1.2 c = 2.5 d = 0.32

Effort

$= a\ (KLOC)^{b}$ *Person-Month*

$= 3.6\ (400)^{1.2}$ *Person-Month*

≈ 4772 *Person-Month*

Development Time

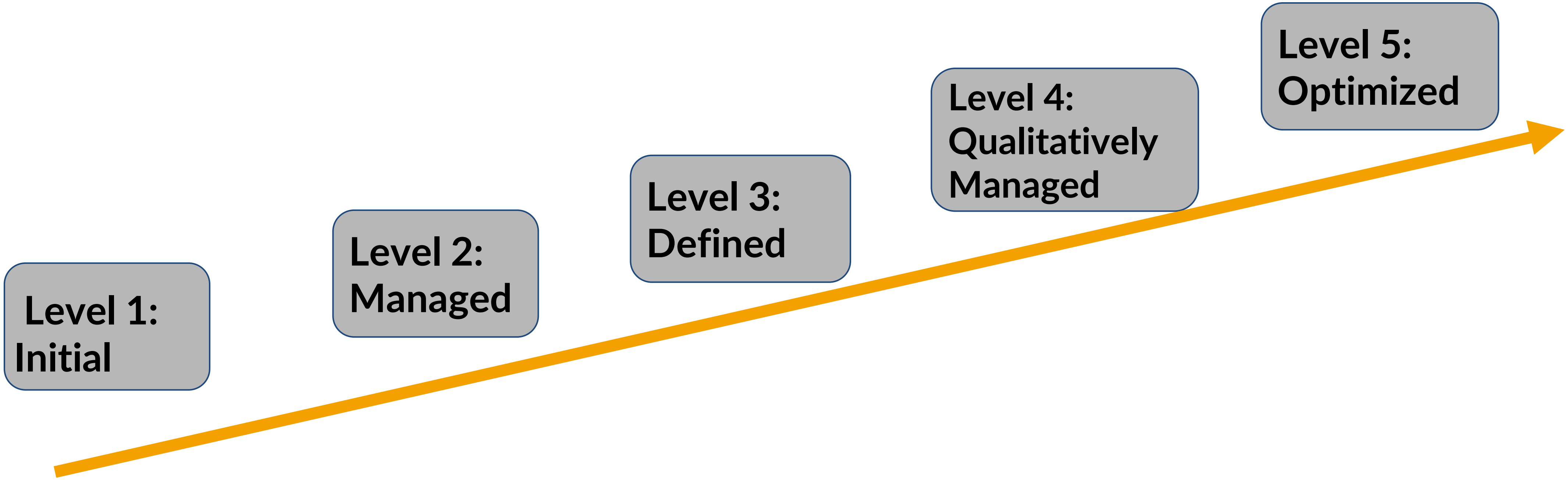
$= c\ (Effort)^{d}$ *Months*

$= 2.5\ (4772)^{0.32}$ *Months*

≈ 38 *Months*

CMMI(Capability Maturity Model Integration)

The CMMI is a process and behavioral model that helps organizations streamline process improvement and encourage productive, efficient behaviors that decrease risks in software, product, and service development.



Level 1: Initial

Level 2: Managed

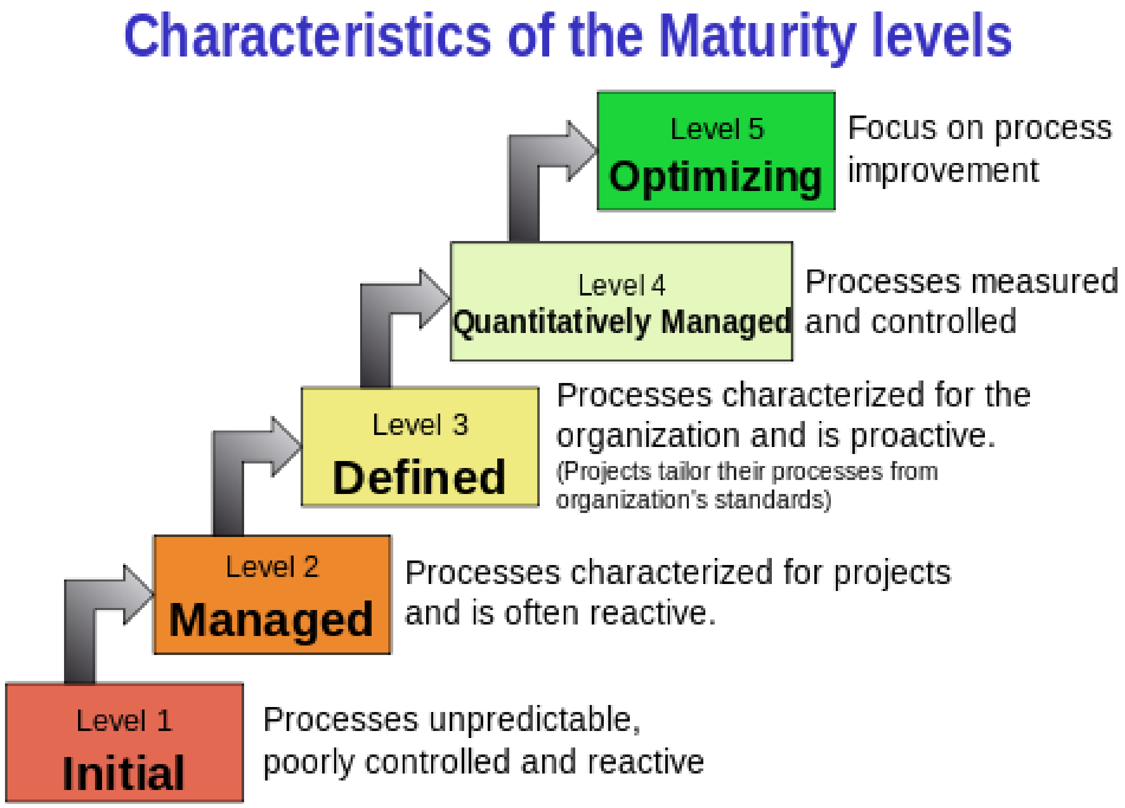
Level 3: Defined

Level 4: Qualitatively Managed

Level 5: Optimized

CMMI Methodology

It's a process improvement approach, aimed at organizational improvement. Agile is an iterative software development methodology, focused on product quality.



Characteristics of the Maturity levels

Level 5 **Optimizing** Focus on process improvement

Level 4 **Quantitatively Managed** Processes measured and controlled

Level 3 **Defined** Processes characterized for the organization and is proactive. (Projects tailor their processes from organization's standards)

Level 2 **Managed** Processes characterized for projects and is often reactive.

Level 1 **Initial** Processes unpredictable, poorly controlled and reactive

THANK YOU