Software Process and Project Metrics

Modified from Roger S. Pressman, Software Engineering:

A Practitioner's Approach 8th Edition, McGraw Hill, 2014

2 types of Metrics

- Productivity Metrics
  - measure of software development output as a function of effort and time applied
- Quality Metrics
  - measure of “the fitness for use” of the work products that are produced

Metrics in the Process and Project Domains

- We use a metric or combination of metrics to provide insight into the software process, project, or the product
- Process Indicators
  - enables an organization to gain insight into the efficacy of software process in order to improve software process
- Project Indicators
  - assess the status of an ongoing project
  - uncover problems areas before they go critical
  - adjust tasks

Process Metrics and Software Process Improvement

- A software process is indirectly measured
- Determine outcomes of the process
  and then derive a set of metrics based on the outcomes
- Examples of outcomes:
  - number of errors uncovered before release
  - defects delivered to and reported end users
  - work product delivered
  - human effort expended
  - calendar time expended
  - schedule conformance

Process Metrics and Software Process Improvement (cont.)

- There are "private and public" uses for process data
- **Private** Metrics:
  - defect rates (by individual), defect rates (by module), errors found during development
- **Public Metrics for the team** members but private to the project
  - defect reported for major software functions, errors found during reviews, LOC or FP per module
- **Public Metrics**
  - defect rates, effort, calendar times

ใช้วัดผลส่วนบุคคล ระดับทีม หรือองค์กร

Project Metrics

- are used by a project manager to adapt project activities
- are collected from **past projects** and used as a basis for the current project
- are used to monitor and control progress as a project proceeds
- examples: production rates (pages of documentation), review hours, FP, delivered source LOC, errors uncovered

เน้นการนำ metrics
มาใช้ monitor and control

2 Categories of Measurement

- Direct Measures
  - process metrics: include cost and effort applied
  - product metrics: LOC, execution speed, memory size
- Indirect Measures
  - process
  - product: functionality, complexity, "…abilities"

ใช้สมการอื่น ๆ

การลงมือดำเนินการเก็บค่า measure คือ measurement ทำได้ทั้งทางตรงและทางอ้อม

Normalization for Metrics

- Normalized data are used to evaluate the process and the product
- Size-oriented normalization - the line of code approach
- Function-oriented normalization - the function point approach

Measure ดิบๆ ที่เก็บมาได้
ควรนำมาทำ normalization ก่อน
เช่น LOC, duration, error points, …etc.

Size-Oriented Metrics

- are direct measures
- Examples
  - KLOC/MM
  - Defects/KLOC
  - Cost = $/LOC
  - documentation = pages of documentation/KLOC
- size-oriented metrics are not accepted because the use of LOC (programming language dependent)

ส่วนใหญ่เริ่มไม่ใช้เพราะ Lang. dependent

Function-Oriented Metrics

- are indirect measures
- focus on functionality
- Examples:
  - FP
  - productivity = FP/MM
  - cost = $/FP
  - errors per FP
  - defects per FP
  - page of documentation per FP

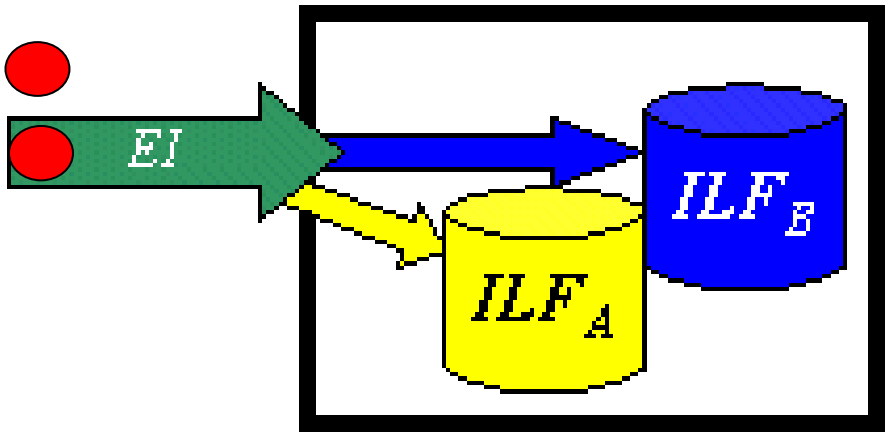
Function-Point Metrics

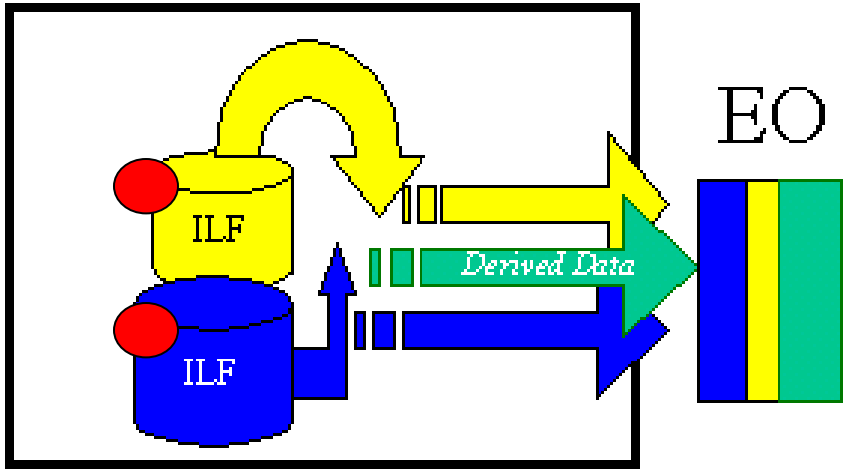
- Function points express the resulting work-product in terms of functionality as seen from the user's perspective
- from users' perspective, a computer system helps them do their job by providing 5 basic functions which is grouped into 2 main functions:
  - Transactional Functions:external inputs, external outputs, external inquiries
  - Data Functions: internal logical files, external interface files

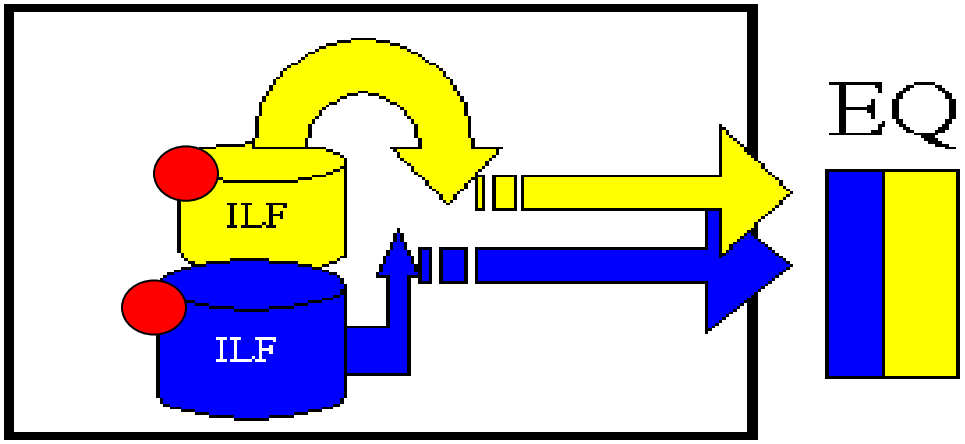
5 System's Basic Functions

- External Input: EI (number of user inputs)
  - each user input that provides distinct application oriented data to the software
- External Outputs: EO (number of user outputs)
  - each user output that provides distinct application oriented data to the software
- External Inquiries: EQ (number of user inquiries)
  - an inquiry is defined as an online input that results in the generation of some immediate software response in the form of an online output
- Internal Logical Files: ILF (number of files)
  - each logical master file is counted
- External Interface Files: EIF (number of external interfaces)
  - the data resides in another system and is maintained by another user or system

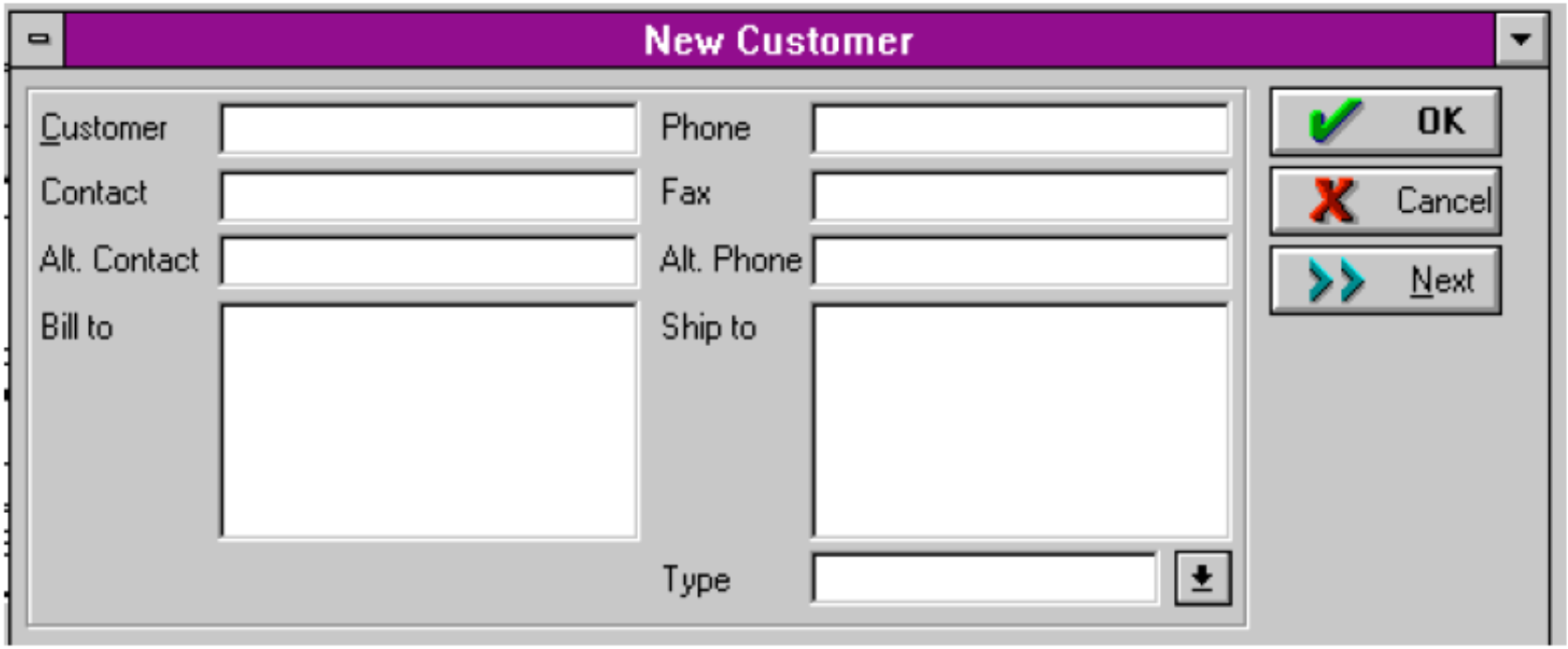
EI, EO, and EQ







An EI Example.



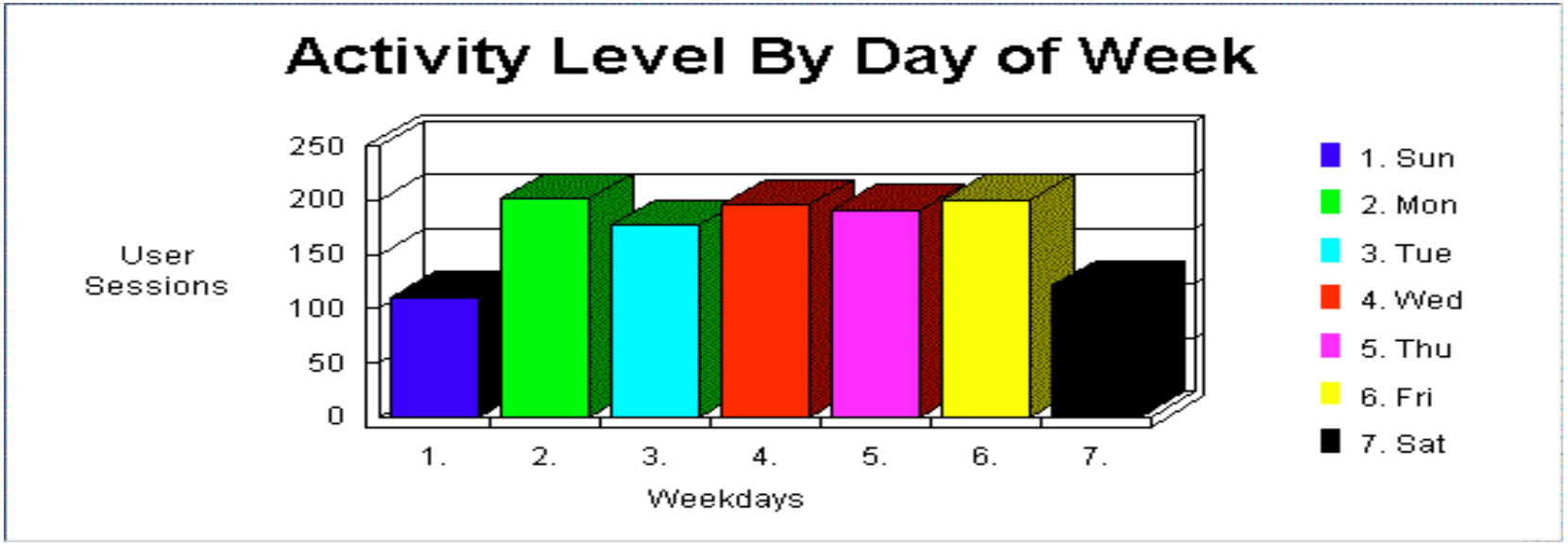
There are 10 data elements in the above screen

1. Customer 2. Phone 3. Contact 4. Fax

5. Alt. Contact 6. Alt. Phone 7. Bill to

8. Ship to 9. Type 10. Buttons

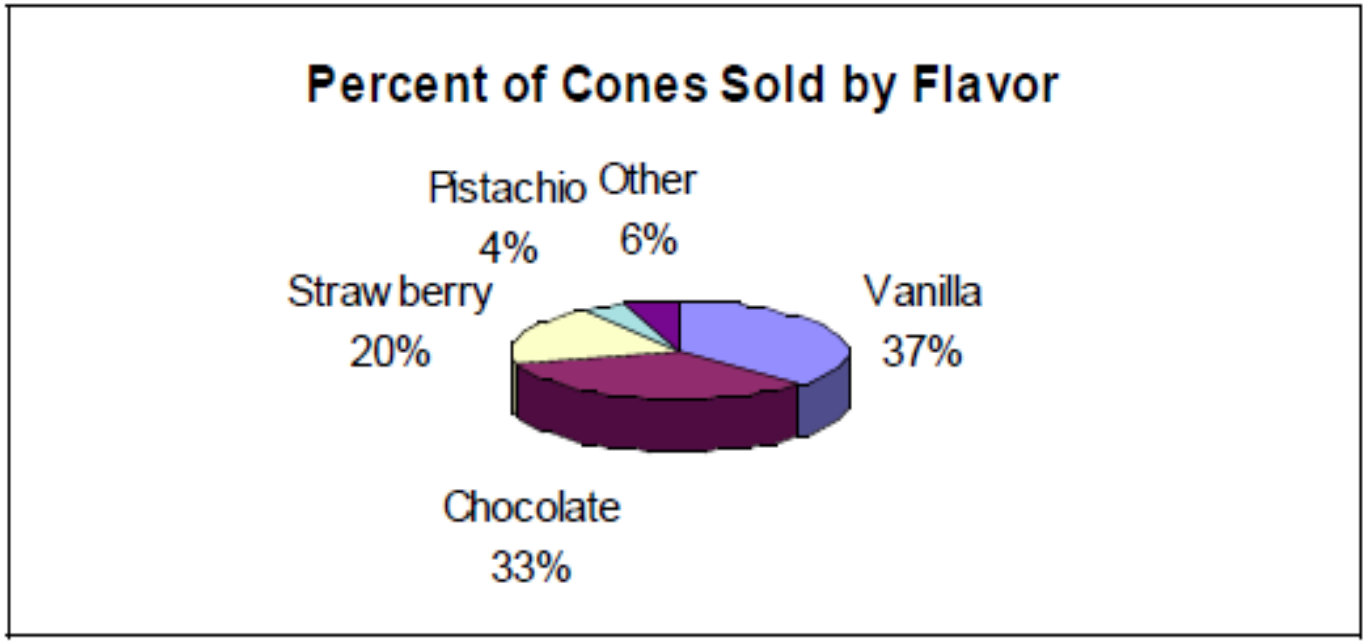
An EO Exercise.



An EO Exercise (cont.)

Flavor	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total
Vanilla	80	85	85	90	110	120	135	145	90	84	75	70	1169
Chocolate	75	80	70	83	100	105	109	120	80	70	69	65	1026
Strawberry	30	35	35	40	70	80	95	105	40	34	25	20	609
Pistachio	8	9	9	9	11	12	14	15	9	8	8	7	119
Other	12	13	13	13	15	17	19	20	14	13	13	12	174
Total	205	222	212	235	306	334	372	405	233	209	190	174	

An EO Exercise (cont.)



An EQ Example.

To begin, select a state from this list or use the map to the right.

Alabama Go

The population of the U.S. is 281,421,906 people (April 1, 2000). The population has grown 13.1% since 1990.

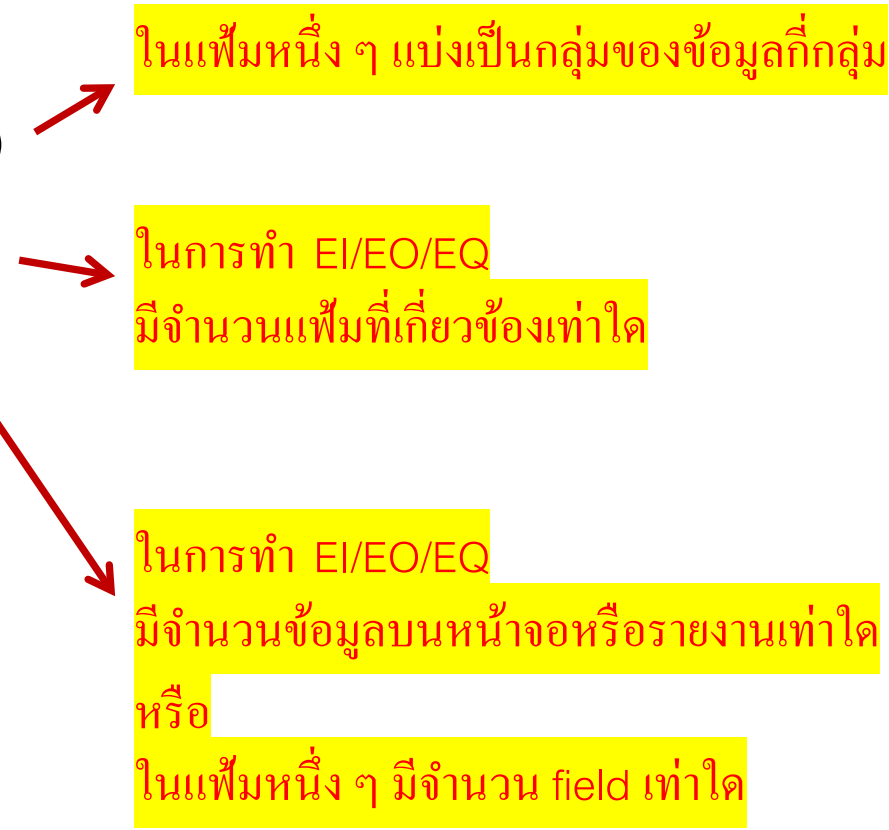
View more USA QuickFacts.

See our thematic maps

Source: U.S. Census Bureau

Determining the Complexity of Basic Functions (if needed)

- Consider more on
  - Record Element Type (RET)
  - File Type Referenced (FTR)
  - Data Element Type (DET)

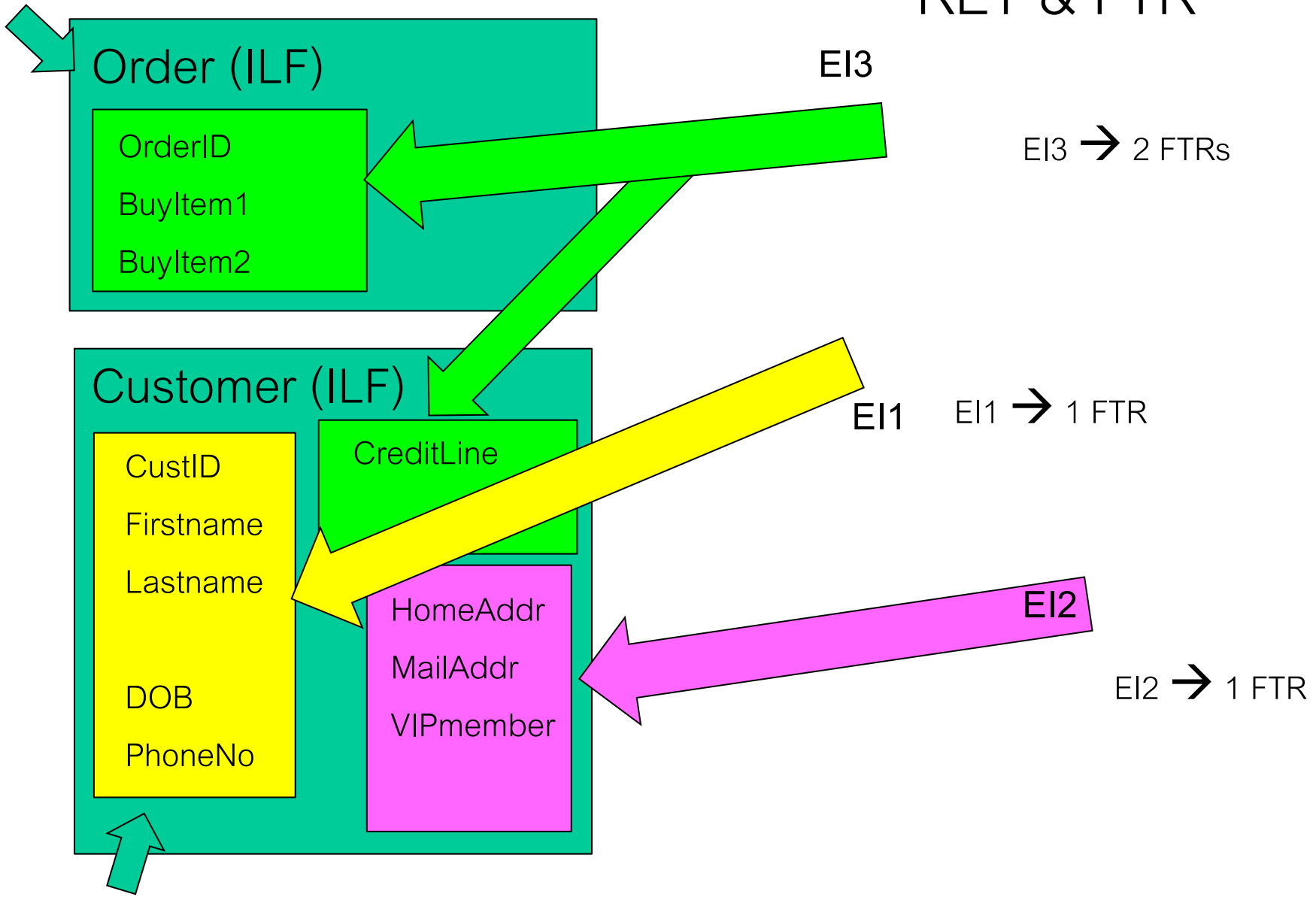


RET, FTR, DET

- *Record Element Type (RET)*:
  - A RET is user recognizable sub group 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)*:
  - A FTR is a file type referenced by a transaction.
  - An FTR must also be an internal logical file or external interface file.

RET & FTR

1 (RETs) Logical groups within Order (ILF)

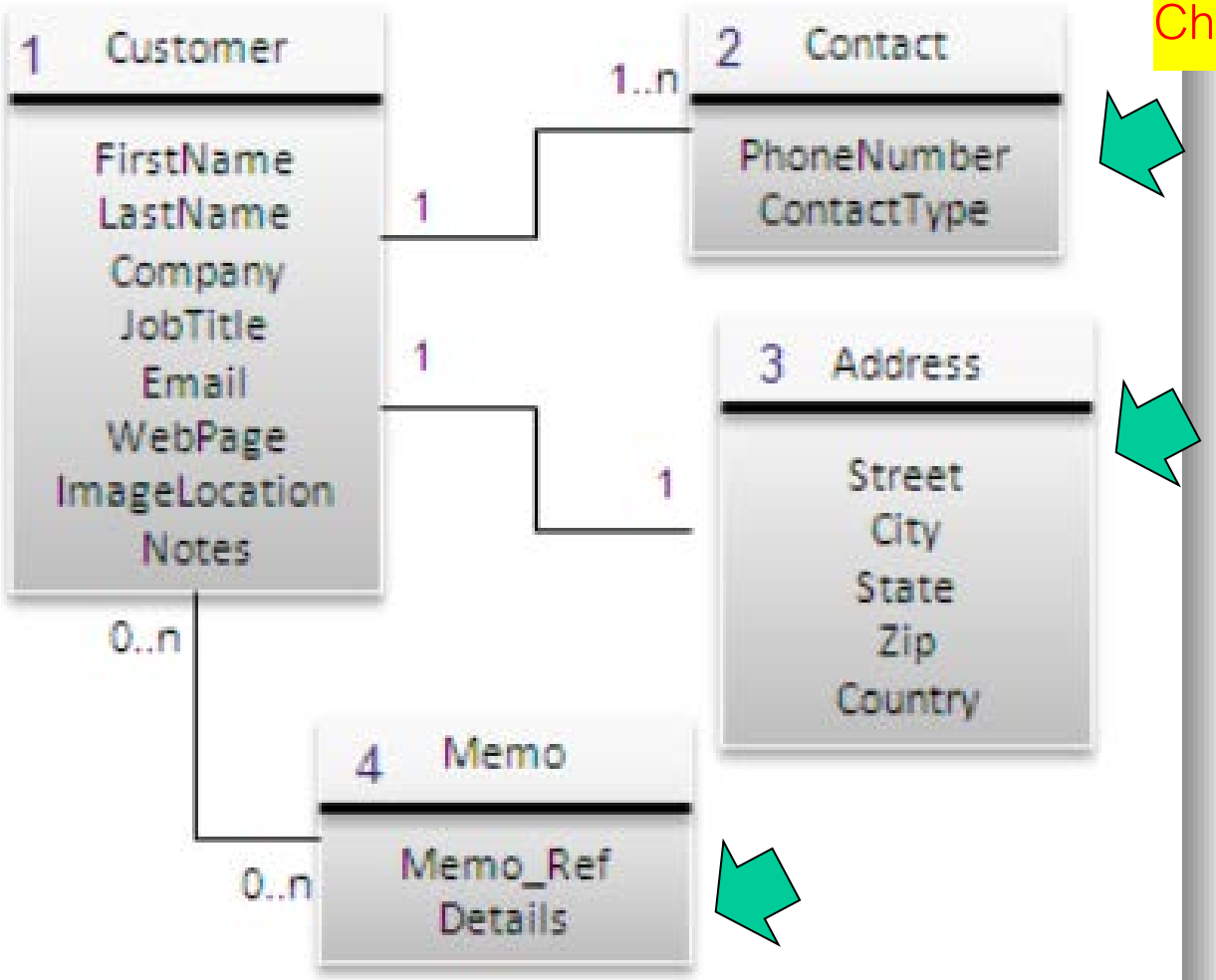


3 (RETs) Logical groups / subgroups within Customer (ILF)

อีกตัวอย่างในการนับ RET คือใช้ Parent-Child Relation

Parent

Child

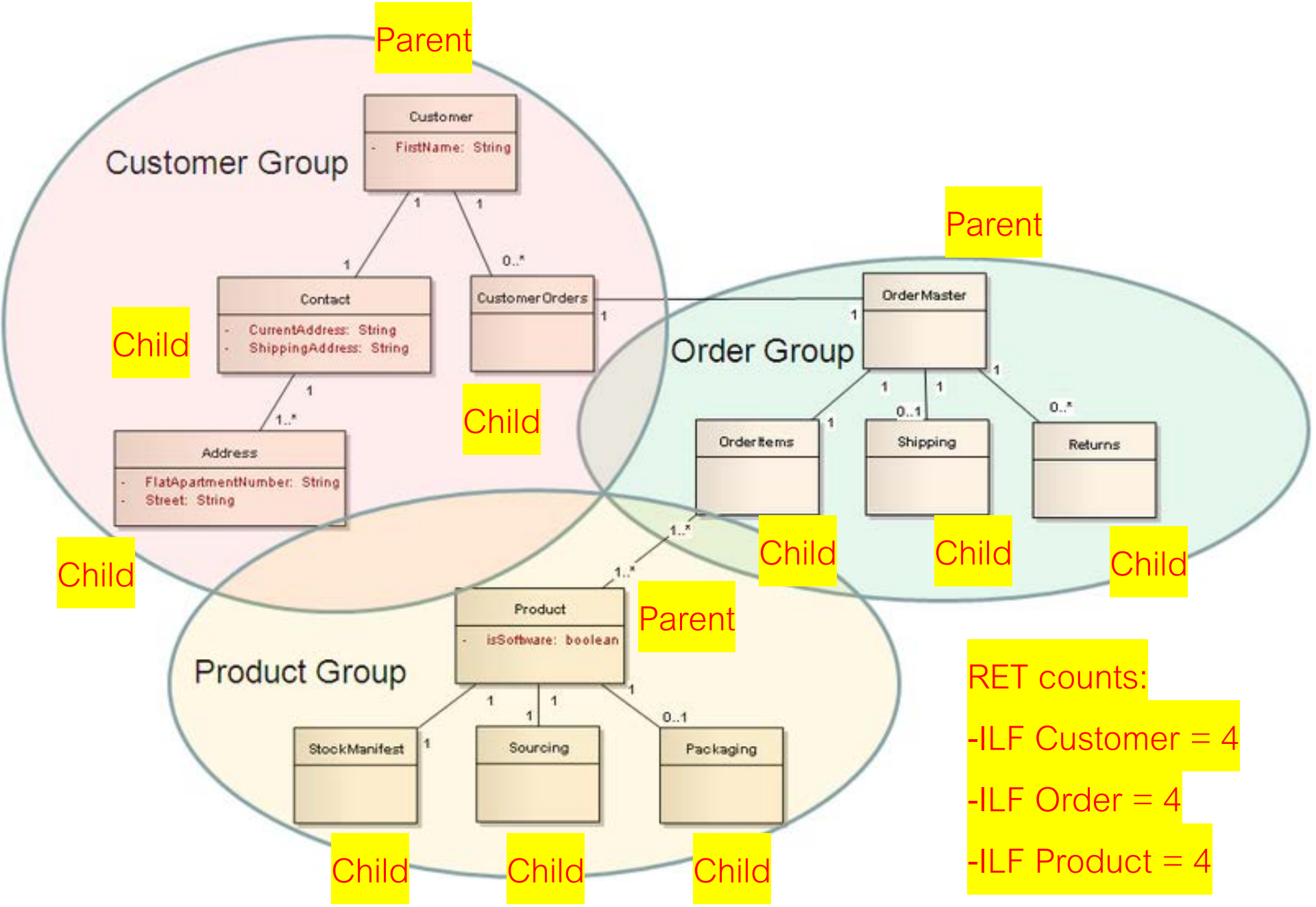


-Parent → 1 RET

-Child → 3 RET

-Total RETs for ILF Customer = 4

https://sourceforge.net/.../Function%20Point%20Analysis/



https://sourceforge.net/.../Function%20Point%20Analysis/

RET, FTR, DET (cont.)

- *Data Element Type (DET)*:
  - A DET is a unique user recognizable, non-recursive (non-repetitive) field.
  - A DET is information that is dynamic and not static. A dynamic field is read from a file or created from DET's contained in a FTR.
  - Additionally, a DET can invoke transactions or can be additional information regarding transactions.
  - If a DET is recursive then only the first occurrence of the DET is considered not every occurrence.

EI Table

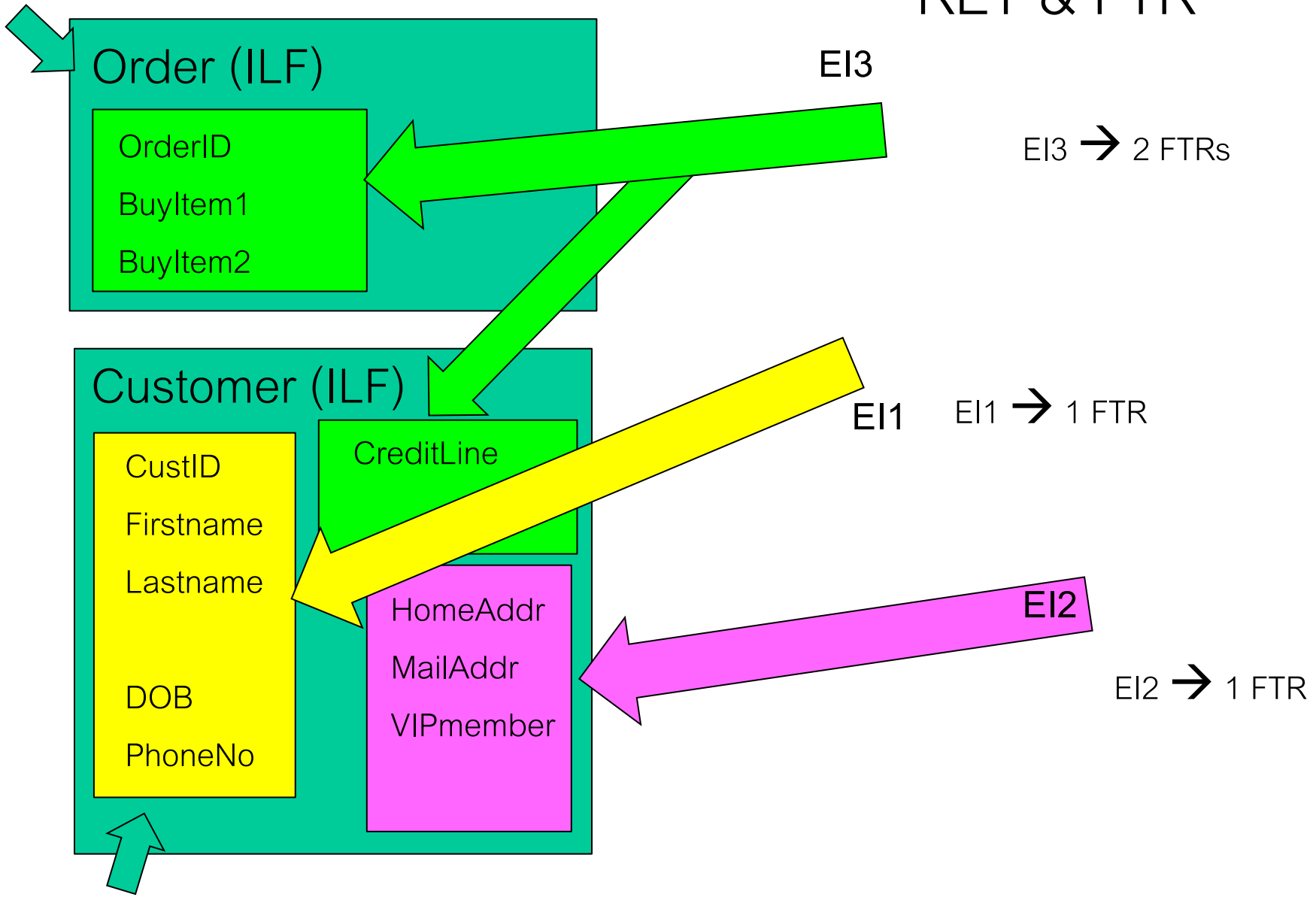
FTR's	DATA ELEMENTS		
	1-4	5-15	> 15
0-1	Low	Low	Ave
2	Low	Ave	High
3 or more	Ave	High	High

File Type Referenced (FTR): A FTR is a file type referenced by a transaction. An FTR must also be an internal logical file or external interface file.

Data Element Type (DET): A DET is a unique user recognizable, non-recursive (non-repetitive) field.

RET & FTR

1 (RETs) Logical groups within Order (ILF)



3 (RETs) Logical groups / subgroups within Customer (ILF)

Shared EO and EQ Table

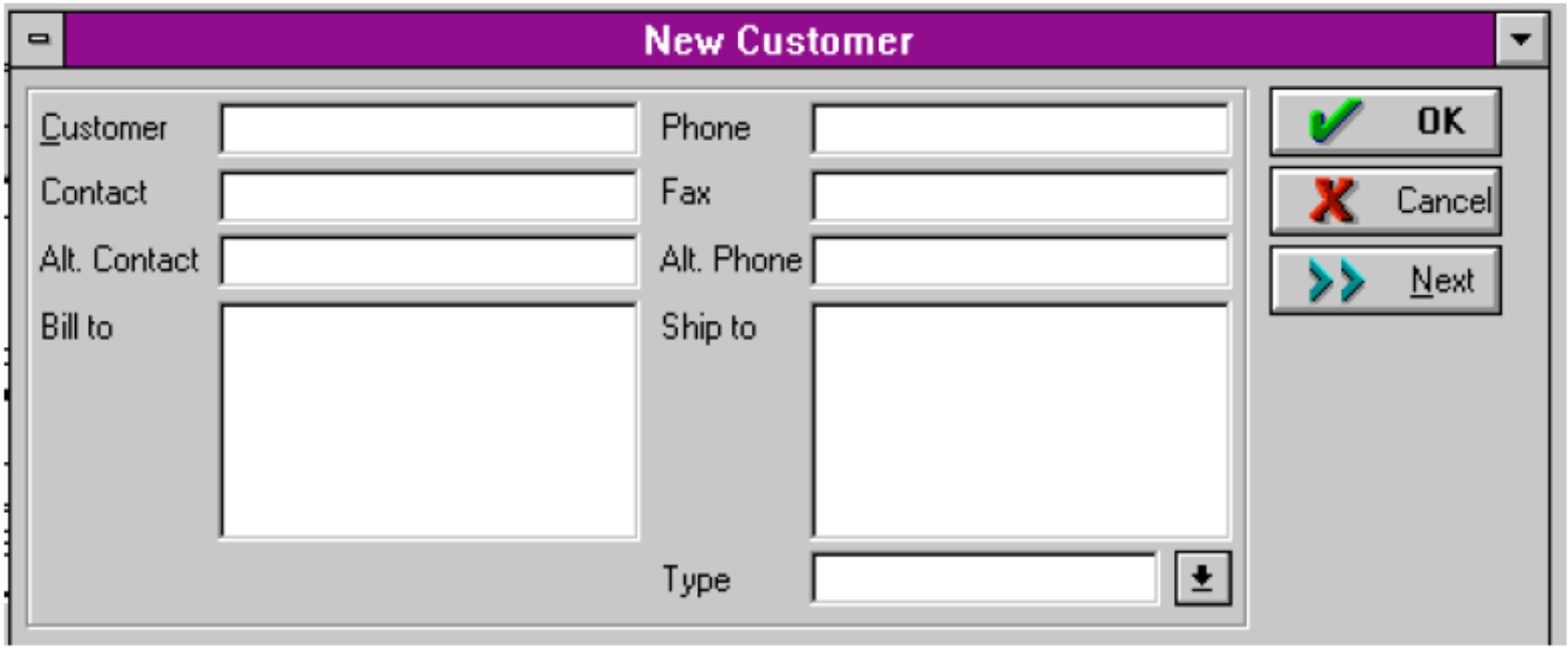
FTR's	DATA ELEMENTS		
	1-5	6-19	> 19
0-1	Low	Low	Ave
2-3	Low	Ave	High
> 3	Ave	High	High

Shared ILF and EIF Table

RET's	DATA ELEMENTS		
	1-19	20 - 50	> 50
1	Low	Low	Ave
2-5	Low	Ave	High
> 5	Ave	High	High

Recap How to count DET's

An EI Example.



There are 10 data elements in the above screen

1. Customer 2. Phone 3. Contact 4. Fax

5. Alt. Contact 6. Alt. Phone 7. Bill to

8. Ship to 9. Type 10. Buttons

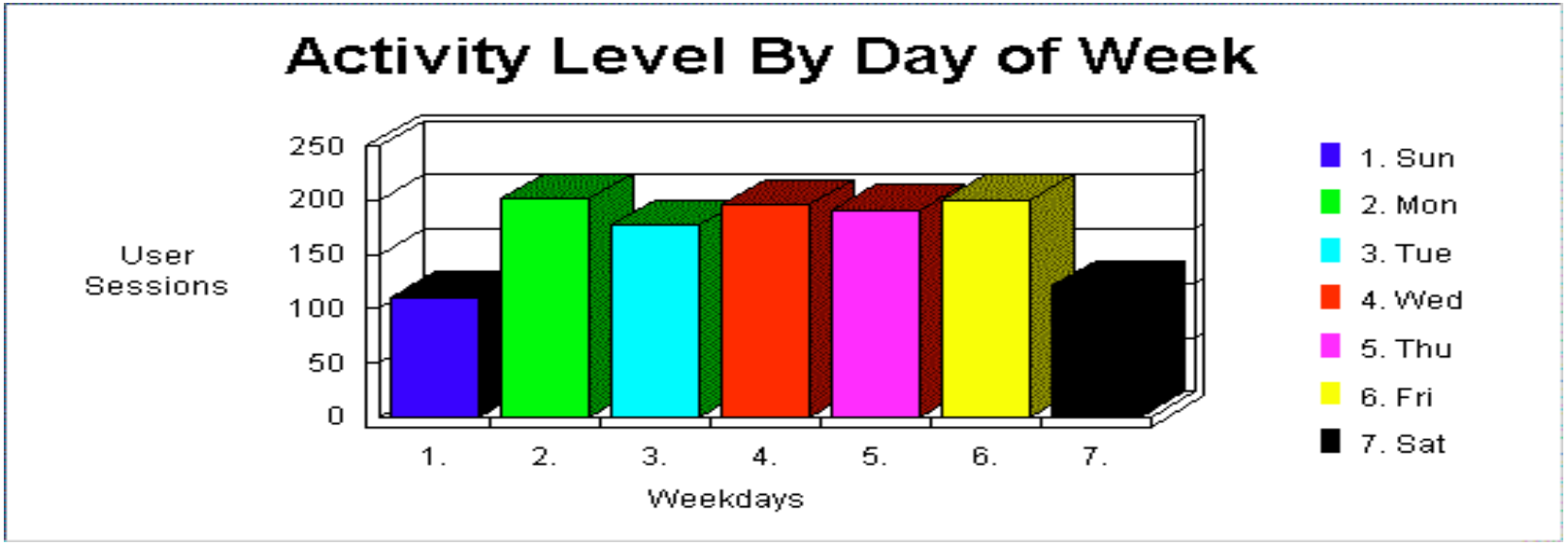
An EO Example.

There are 10 data elements in the following table

1. Days
2. Hits
3. % of Total Hits
4. User Sessions
5. Total Hits (weekday)
6. Total % (weekday)
7. Total User Sessions (weekday)
8. Total Hits (weekend)
9. Total % (weekend)
10. Total User Sessions (weekend)

Activity Level by Day of the Week				
	Day	Hits	% of Total Hits	User Sessions
1	Sun	1004	8.73%	111
2	Mon	1887	16.41%	201
3	Tue	1547	13.45%	177
4	Wed	1975	17.17%	195
5	Thu	1591	13.83%	191
6	Fri	2209	19.21%	200
7	Sat	1286	11.18%	121
	Total Weekdays	**9209**	**80.08%**	**964**
	Total Weekend	**2290**	**19.91%**	**232**

An EO Exercise.



The same data could be processed and presented as bar graph. But on the following bar graph there are only two data elements (user session and day of week). The bar graph is a separate

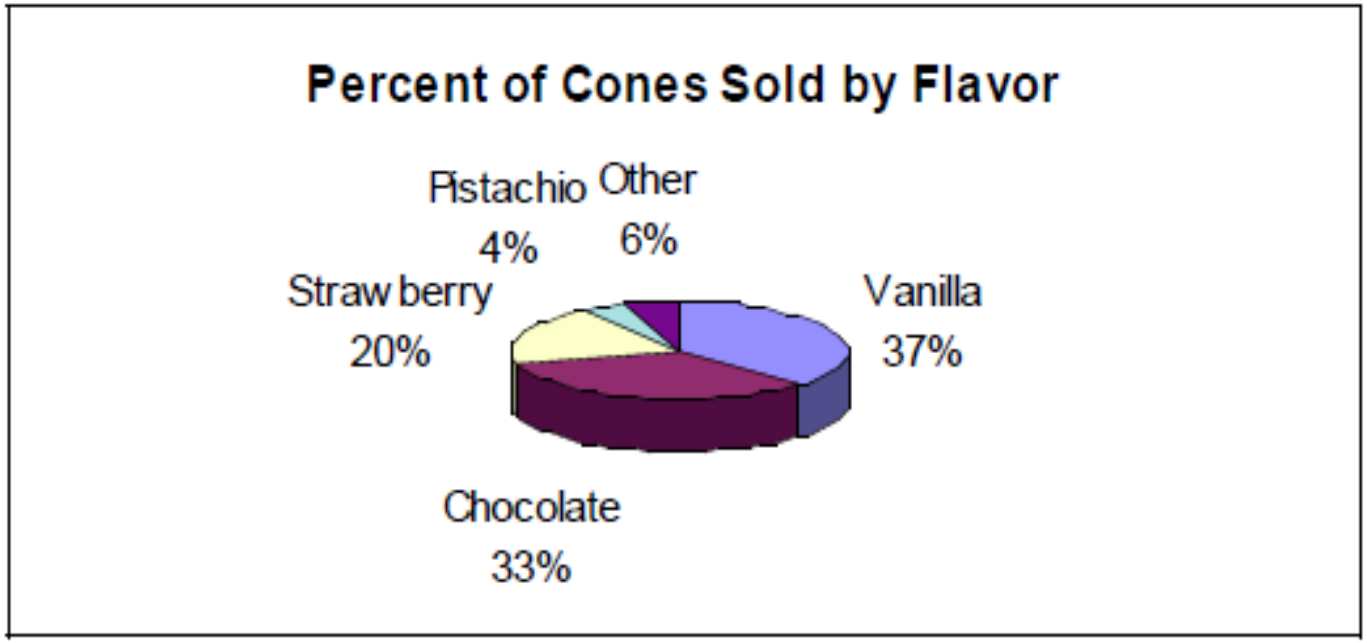
An EO Exercise (cont.)

Flavor	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total
Vanilla	80	85	85	90	110	120	135	145	90	84	75	70	1169
Chocolate	75	80	70	83	100	105	109	120	80	70	69	65	1026
Strawberry	30	35	35	40	70	80	95	105	40	34	25	20	609
Pistachio	8	9	9	9	11	12	14	15	9	8	8	7	119
Other	12	13	13	13	15	17	19	20	14	13	13	12	174
Total	205	222	212	235	306	334	372	405	233	209	190	174	

1. How many data elements are there in the above chart?

Flavor, Month, Total by Flavor, Total by Month or 4 DET's

An EO Exercise (cont.)



Flavor and Percent or 2 DET's

Computing Function Points Metrics

Measurement parameter	count		Weighting Factor simple		average		complex		
number of user inputs		X	3		4		6	=	
number of user outputs		X	4		5		7	=	
number of user inquiries		X	3		4		6	=	
number of files		X	7		10		15	=	
number of external interfaces		X	5		7		10	=	
count total	→								

Computing Function Points Metrics

- count 1 system's basic functions
- associated a complexity value with each count
- the determination of complexity is somewhat subjective
- compute function point

$$FP = \text{count-total } X \left[0.65 + 0.01 X \sum F_i\right]$$

- F_i (i = 1 to 14) are complexity adjustment values which are constant values: 0 = irrelevant, 1 = incidental, 2 = moderate, 3 = average, 4 = significant, 5 = essential

Technical Complexity Factors

- Data Communication
  - the data and control information used in the application are sent or received over communication facilities
- Distributed Data Processing
  - distributed or processing functions are a characteristic of the application within the application boundary
- Performance
  - in either response or throughput, influence in the design, development, installation, and support of the application

Technical Complexity Factors (cont.)

- Heavily-Used Configuration
  - a heavily used operational configuration, requiring special design consideration
- Transaction Rate
  - is high and influences the design, development, installation, and support
- On-line Data Entry
- End-User Efficiency
  - the on-line functions provided emphasize a design for end-user efficiency
- On-line update
  - for internal logical files

Technical Complexity Factors

- Complexity Processing
- Reusability
  - is developed to be usable in other applications
- Installation Ease
  - conversion and installation ease are characteristics of the application
- Operational Ease (Reliable Backup and Recovery)
  - effective startup, backup, and recovery procedures were provided and tested during the system phases
- Multiple Sites:
  - to be developed for multiple sites or multiple organization
- Facilitate Change

FP Table

Type of Component	Complexity of 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 Number of Unadjusted Function Points		___
		Multiplied Value Adjustment Factor		___
		Total Adjusted Function Points		___

(1) Consider a project with the following functional units

- Number of user inputs = 50
- Number of user outputs = 40
- Number of user enquiries = 35
- Number of user files = 06
- Number of external interfaces = 04

Assuming all weighing factors and complexity adjustment factors and as average, the function points for the project will be?

1. Typical complexity averages are as follows:

TABLE 1: Function point complexity weights.

Measurement parameter	Weighting fa		
	Simple	Average	Complex
Number of user inputs	3	4	6
Number of user outputs	4	5	7
Number of user inquiries	3	4	6
Number of files	7	10	15
Number of external interfaces	5	7	10

- Number of user inputs = 50
- Number of user outputs = 40
- Number of user enquiries = 35
- Number of user files = 06
- Number of external interfaces = 04

AVERAGE complexity weights = `{4, 5, 4, 10, 7}` for the 5 complexities respectively.

Total counts =

50*4 + 40*5 + 35*4 + 6*10 + 4*7 = 200 + 200 + 140 + 60 + 28

Hence, Total counts = 628

2. Typical Characteristic weights are as follows:

AVERAGE characteristic weight = 3.

$$FP = \text{count-total} \times [0.65 + 0.01 \times \sum F_i]$$

3. Function point = `FP = UPF x VAF`

`UFP` = Sum of all the complexities i.e. the 5 parameters provided in the question,
`VAF` = Value added Factor i.e. `0.65 + (0.01 * TDI)`,
`TDI` = Total Degree of Influence of the 14 General System Characteristics.

Measurement parameter	Count	Weighting Factor(Average)	Total
Number of user inputs	50	4	=200
Number of user outputs	40	5	=200
Number of user enquiries	35	4	=140
Number of user files	6	10	=60
Number of external interfaces	4	7	=28

Thus function points can be calculated as:

```
(200 + 200 + 140 + 60 + 28) x (0.65 + (.01 x (14 x 3)))
= 628 x (0.65 + 0.42)
= 628 x (1.07)
= 672
```

What if $\sum F < 0.35$

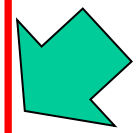
Thus the function points for the project will be **672**.

(2) Consider a project with the following functional units

- Number of user inputs = 50
- Number of user outputs = 40
- Number of user enquiries = 35
- Number of user files = 06
- Number of external interfaces = 04

2.1 If the weighing factors are the following:

user inputs => 20 (L), 20 (A), 10 (H)
user outputs =>20 (L), 15 (A), 5 (H)
user enquiries => 20 (L), 10 (A), 5 (H)
user files -> 5(L), 1(A)
external interfaces -> 3(L), 1(A)



2.2 If the system needs to be

(1) highly reusable
(2) distributed data processing

All the remaining factors just be average, what he function points for the project will be?

FP Table

Type of Component	Complexity of Components			
	Low	Average	High	Total
External Inputs	20 x 3 = ___	20 x 4 = ___	10 x 6 = ___	200
External Outputs	20 x 4 = ___	15 x 5 = ___	5 x 7 = ___	190
External Inquiries	20 x 3 = ___	10 x 4 = ___	5 x 6 = ___	130
Internal Logical Files	5 x 7 = ___	1 x 10 = ___	0 x 15 = ___	45
External Interface Files	3 x 5 = ___	1 x 7 = ___	0 x 10 = ___	22

Total Number of Unadjusted Function Points	587
Multiplied Value Adjustment Factor	1.11
Total Adjusted Function Points	651.57

12*3 + 2*5 = 46 => [0.65 + 0.01* 46]

2 factors rated = score 5 (high) and

12 factors rated = score 3 (avg)

LOC and FP

- Relationship between LOC and FP depends on the programming language used

Programming Language	*LOC/FP (average)*
assembly	*320*
C	*128*
COBOL	*106*
FORTRAN	*106*
Pascal	*90*
C++	*64*
Ada95	*53*
Visual Basic	*32*
Smalltalk	*22*
Powerbuilder (code generator)	*16*
SQL	*12*

Metrics for Software Quality

- 3 Viewpoint of McCall's quality Factors
  - Product operation (using it): correctness, reliability, usability, integrity, efficiency
  - Product revision (changing it): maintainability, flexibility, testability
  - Product transition (modify it to work in a different environment): portability, reusability, interoperability

Definitions and Measures of Some Quality Metrics

- Correctness: the degree to which a program operates according to specification
  - defects/KLOC
  - a defect is a verified lack of conformance to requirements
- Maintainability: the degree to which a program amenable to change (correct, adapt, or enhance)
  - indirect measure:
    - MTTC (Mean Time To Change)
    - MTTR (Mean time to repair) ก็ได้

Maintainability is the ease with which a program can be correct/modified

Definitions and Measures of Some Quality Metrics (cont.)

- Integrity: the degree to which a program is impervious to outside attack
  - integrity = $\sum$ [1 – (threat × (1 - security))] (สูตรถูกต้องแล้ว)
  - threat is the probability that an attack of a specific type will occur within a given time
  - security is the probability that the attack of a specific type will be repelled
  - เช่น Threat1 = 0.25 คือโอกาสในการเกิดการโจมตีจาก Thread1 เป็น 0.25 และมี Security1 = 0.95 คือโอกาสในการต้านทางการโจมตีเป็น 0.95 แล้ว

    Integrity = [1-(0.25 x (1-0.95)] = 0.98 สำหรับ Thread1
- Usability: the degree to which a program is easy to use
  - skill required to learn the system
  - time required to become moderately efficient in the use of the system
  - the net increase in productivity
  - a subjective assessment

Defect Removal Efficiency

- DRE = E/(E +D)
- E is the number of errors found before delivery of the software to end-user
- D is the number of defects found after delivery
- Ideal value for DRE is 1

Availability

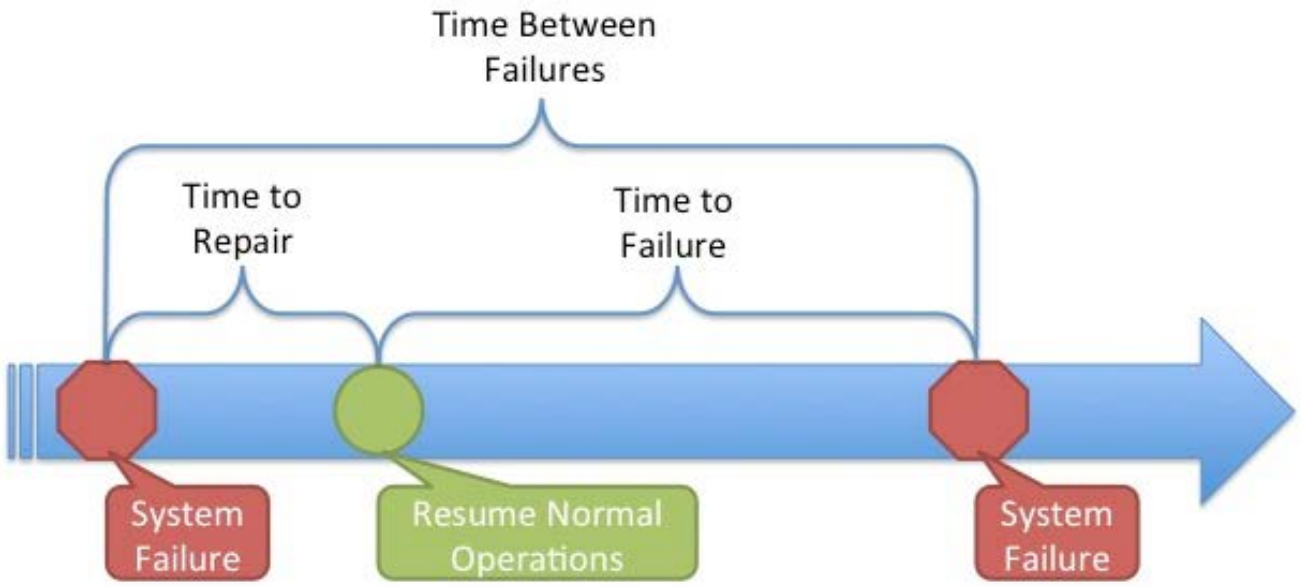
- Availability: *Software availability is the probability that a program is operating according to requirements* at a given point in time and is defined as

$$\text{Availability} = [\text{MTTF}/(\text{MTTF} + \text{MTTR})] \ 100\%$$

ถ้า MTTR น้อย เช่น 1 วัน และ MTTF มาก เช่น 300 วัน แล้ว

Availability = [300/301] 100% เกือบ 100% ดีมาก

Reliability วัดโดย MTBF = MTTR + MTTF



- MTBF (Mean Time Between Failure)
- MTTF (Mean Time To Failure)
- MTTR (Mean Time To Repair)

แบบง่าย
โดย assume ว่า MTTR ไม่มาก

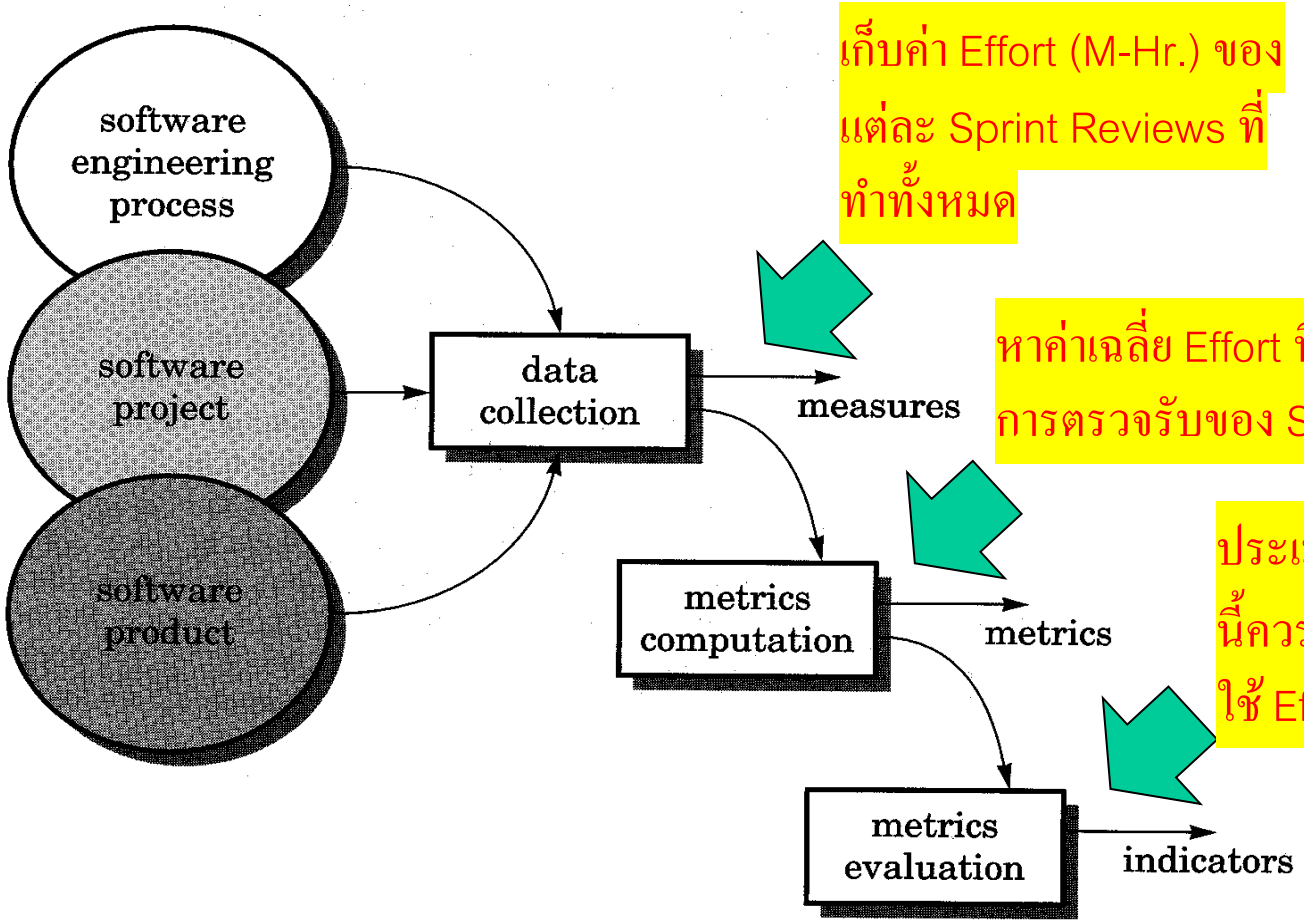
สรุป

- **Correctness**
  - **Defects per KLOC**
- **Maintainability**
  - **Mean time to change (MTTC)**
  - **Mean time to repair (MTTR)**
- **Integrity**
  - **Threat and security**
  - **integrity = Σ [1 – (threat $\times$ (1 - security))]**
- **Usability**
  - **Amount of Time It Takes to Learn to Use the Software**
- **Availability**
  - **Availability = [MTTF/(MTTF + MTTR)] 100%**
- **Reliability**
  - **MTBF = MTTR + MTTF**

Integrating Metrics Within The Software Process

- Why is it important to measure the software process and product?
- The current status must be known, in order to improve
- Measurements establish a project baseline
- Baseline serves as a basis for estimation
- Data are collected in order to establish a baseline
- after measures, metrics are computed
- metric evaluation produces a set of indicators that guide the project or process

Software Metrics Collection Process



เก็บค่า Effort (M-Hr.) ของแต่ละ Sprint Reviews ที่ทำทั้งหมด

หาค่าเฉลี่ย Effort ที่ใช้ในการตรวจรับของ SCRUM project นี้

ประเมินว่า SCRUM project นี้ควรปรับปรุงใหม่ เพราะใช้ Effort มากเกิน Basis line

โจทย์

1. จงคำนวณหาค่า FP ที่มีการปรับแต่งด้วย Complexity Adjustment Values เรียบร้อยแล้วของระบบ Online Shopping ซึ่งมีลักษณะดังต่อไปนี้

Basic Function	Count	#DET	#FTR	#RET
EI	2	1	0	n/a
EI	3	8	2	n/a
EI	1	15	5	n/a
EI	1	18	6	n/a
EQ	1	5	4	n/a
EQ	2	19	2	n/a
EQ	1	8	1	n/a
EO	1	8	4	n/a
EO	2	20	2	n/a
ILF	2	25	n/a	2
ILF	1	3	n/a	1
EIF	2	13	n/a	3

จงแสดงวิธีทำ

2. กำหนดให้ Metric เฉลี่ยของทีมงานซึ่งมีอยู่ 5 คน คือ

- Productivity คือ 10 FP/MM
- Cost คือ 2000 Baht/FP
- Doc คือ 100 Pages/FP
- Doc Cost คือ 5 Baht/Page
- เงินเดือนสมาชิกในทีมเฉลี่ยอยู่ที่ 20050 บาท

2.1 จงคำนวณหาค่าใช้จ่ายทั้งสิ้นที่ต้องลงทุนกับการพัฒนาซอฟต์แวร์ Online Shopping นี้

2.2 ถ้าทุกคนในทีมช่วยกัน งานจะเสร็จภายในระยะเวลาเท่าใด

2.3 ถ้าใช้ภาษา C++ ในการเขียนโปรแกรม จะต้องเขียนโปรแกรมกี่ KLOC

Reference Card

External Inputs (EI)

Files Type Referenced (FTR)	Data Elements		
	1-4	5-15	Greater than 15
Less than 2	Low (3)	Low (3)	Average (4)
2	Low (3)	Average (4)	High (6)
Greater than 2	Average (4)	High (6)	High (6)

External Outputs (EO)

File Types Referenced (FTR)	Data Elements		
	1-5	6-19	Greater than 19
less than 2	Low (4)	Low (4)	Average (5)
2 or 3	Low (4)	Average (5)	High (7)
Greater than 3	Average (5)	High (7)	High (7)

External Inquiry (EQ)

File Types Referenced (FTR)	Data Elements		
	1-5	6-19	Greater than 19
less than 2	Low (3)	Low (3)	Average (4)
2 or 3	Low (3)	Average (4)	High (6)
Greater than 3	Average (4)	High (6)	High (6)

Internal Logical Files (ILF)

Record Element Types (RET)	Data Elements		
	1 to 19	20 - 50	51 or More
1 RET	Low (7)	Low(7)	Average (10)
2 to 5 RET	Low (7)	Average (10)	High (15)
6 or More RET	Average (10)	High (15)	High (15)

External Interface File (EIF)

Record Element Types (RET)	Data Elements		
	1 to 19	20 - 50	51 or More
1 RET	Low (5)	Low(5)	Average (7)
2 to 5 RET	Low (5)	Average (7)	High (10)
6 or More RET	Average (7)	High (10)	High (10)