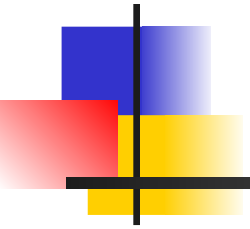
SOEN 384
Management, Measurement and Quality Control

http://users.encs.concordia.ca/~s384_2/



Instructor: Dr. Olga Ormandjieva

Lectures 11&12:

Management Key Area (KA) 1: Initiation and scope definition

Functional Size Measurement & COSMIC

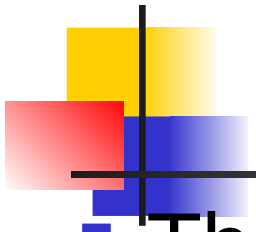
1. COSMIC Measurement Method ISO 19761
2. COSMIC Method v3.0.1 Measurement Manual
3. Villavicencio & Abran. A framework for education in software measurement Evaluation version - 08/2013

Agenda

- **Review**
- Management Key Area (KA) 1: Initiation and scope definition
- MEASURES FOR THE REQUIREMENTS PHASE: Functional Size Measurement (FSM)
- Exercises
- Next?

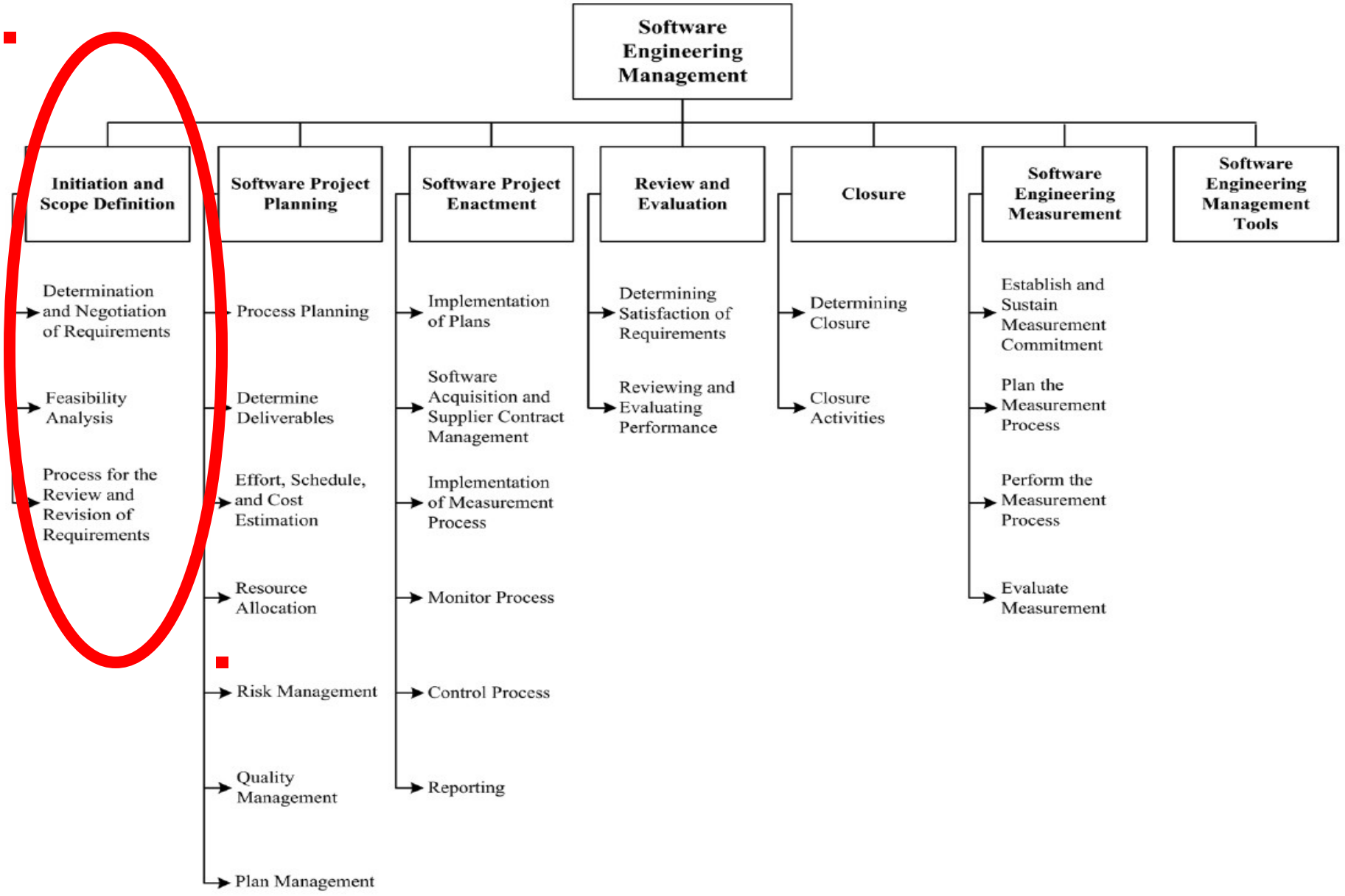
Project managers' goal:
To delivering projects on time, within budget and according to scope and quality.

Software Scope Definition (Pressman, Section 31.3.1)



- The first software management activity is the determination of software scope:
  - Context
  - Information Objectives
  - Function and performance
- A statement of software scope must be bounded
  - Testable FRs and NFRs
  - Constraints

SEM KA


Software Engineering Management
Initiation and Scope Definition
Determination and Negotiation of Requirements
Feasibility Analysis
Process for the Review and Revision of Requirements
Software Project Planning
Process Planning
Determine Deliverables
Effort, Schedule, and Cost Estimation
Resource Allocation
Risk Management
Quality Management
Plan Management
Software Project Enactment
Implementation of Plans
Software Acquisition and Supplier Contract Management
Implementation of Measurement Process
Monitor Process
Control Process
Reporting
Review and Evaluation
Determining Satisfaction of Requirements
Reviewing and Evaluating Performance
Closure
Determining Closure
Closure Activities
Software Engineering Measurement
Establish and Sustain Measurement Commitment
Plan the Measurement Process
Perform the Measurement Process
Evaluate Measurement
Software Engineering Management Tools

Initiation and Scope Definition

Deals with the decision to embark on a software engineering project

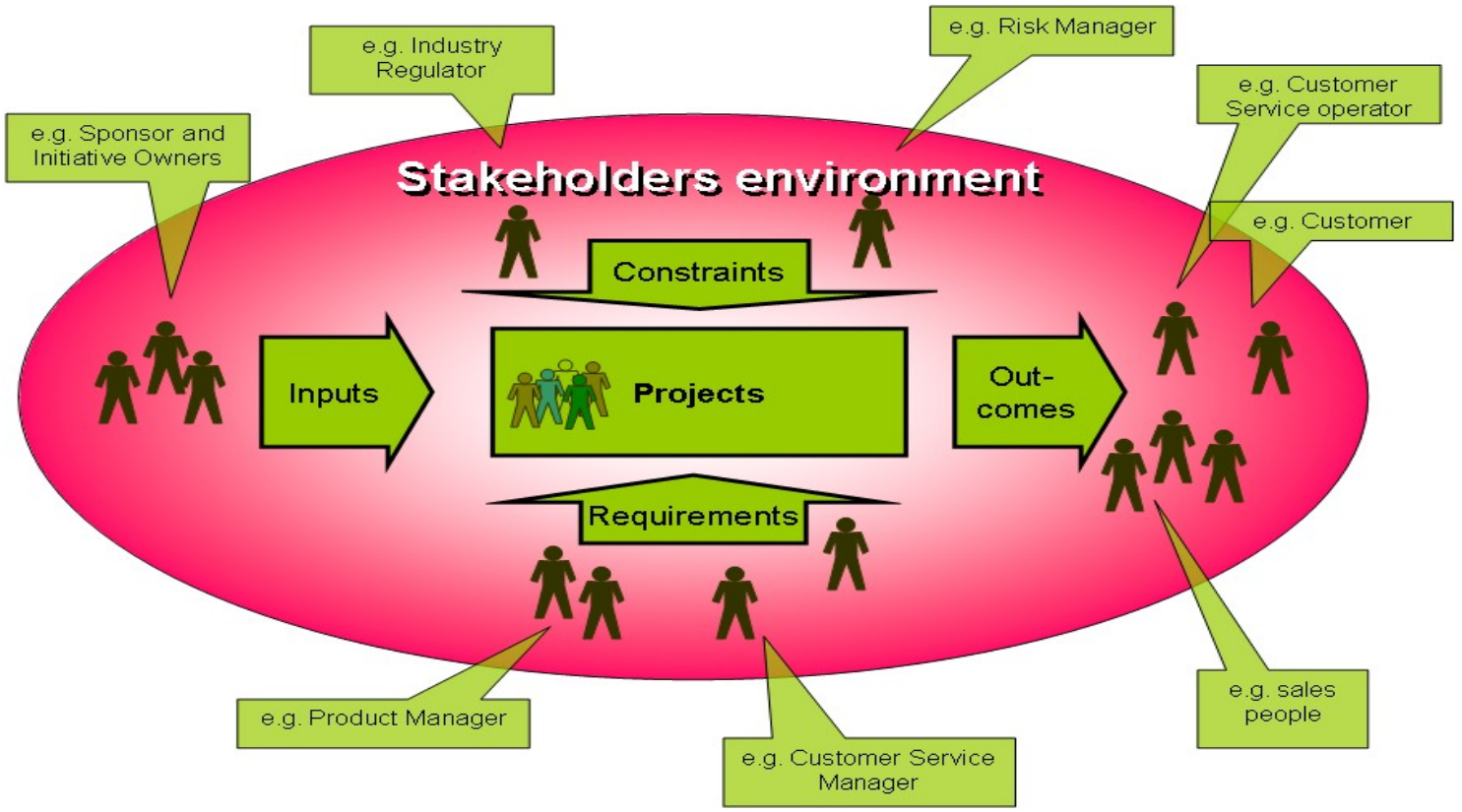
1.1. Determination and Negotiation of Requirements

1.2. Feasibility Analysis

1.3. Process for the Review and Revision of Requirements

SEM – KA 1: Initiation and Scope Definition

- 1.1. Determination and Negotiation of R



Initiation and Scope Definition

- *1.1. Determination and Negotiation of Requirements*
  - Activities: requirements elicitation, analysis, specification, and validation
  - Outcome: determination of project scope in order to meet objectives and satisfy constraints

SEM – KA 1: Initiation and Scope Definition

- **1.2. Feasibility Analysis**
  - Ensures that adequate capability and resources are available in the form of people, expertise, facilities, infrastructure, and support (either internally or externally) to ensure that the project can be successfully completed in a

    timely and cost-effective manner

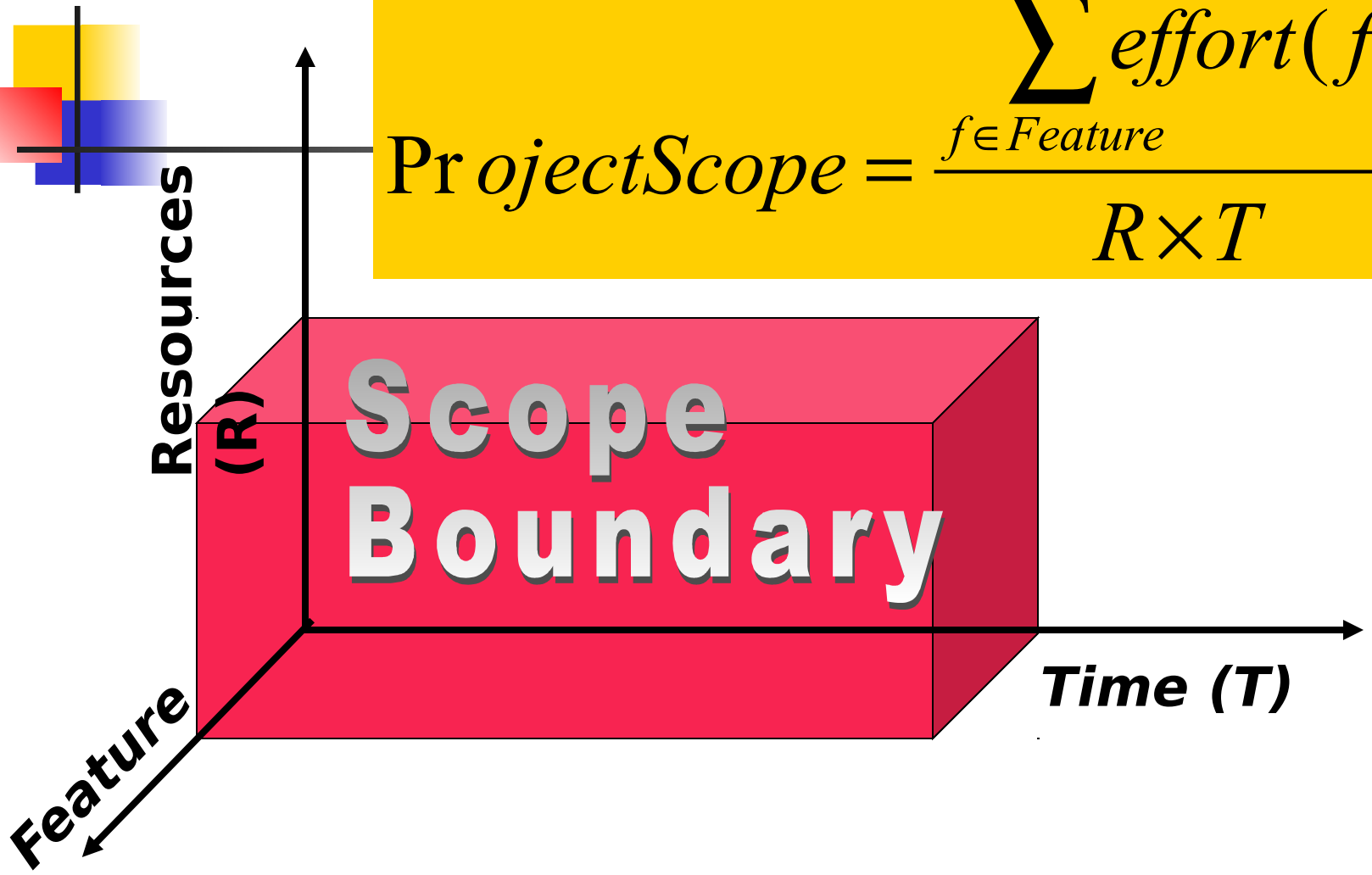
Initiation and Scope Definition

1.2. Feasibility Analysis

- Activities: description of project objectives, evaluation of alternative approaches, approximate e
- Outcomes: An initial project and product scope statement, project deliverables, project duration

Based on functional size of requirements (NEXT)

$$\mathrm{Pr}\,ojectScope = \frac{\sum_{f \in Feature} effort(f)}{R \times T}$$



SEM – KA1: Initiation and Scope Definition

- **1.3. Process for the Review and Revision of Requirements**
  - Inner consistency checking of all related documents
  - Verifying against defined best practices (rules, checklists)
  - One efficient way to provide feedback on the author's work (Inspections and peer reviews)
  - Plan for a managed-change approach
    - Change analysis, traceability analysis and risk analysis

Initiation and Scope Definition

1.3. Process for the Review and Revision of Requirements

- Outcomes: requirements inspection procedures, change management procedures, iterative cycle retrospectives

Agenda

- Review
- Management Key Area (KA) 1: Initiation and scope definition
- **MEASURES FOR THE REQUIREMENTS PHASE: Functional Size Measurement**
- Exercises
- Next?

MEASURES FOR THE REQUIREMENTS PHASE: Functional Size Measurement (FSM)

- What is functional size?

Functional size is defined in ISO 14143-1:2007(E) as: "a size of the software derived by quantifying the Functional User Requirements (FUR)," where FUR is as "a sub-set of the User Requirements describing what the software shall do, in terms of tasks and services"

- Examples of FUR are:
  - input of students data in a registration system;
  - calculate the average mark of students in a course;
  - list the students that are above the average.

MEASURES FOR THE REQUIREMENTS PHASE: Functional Size Measurement (FSM)

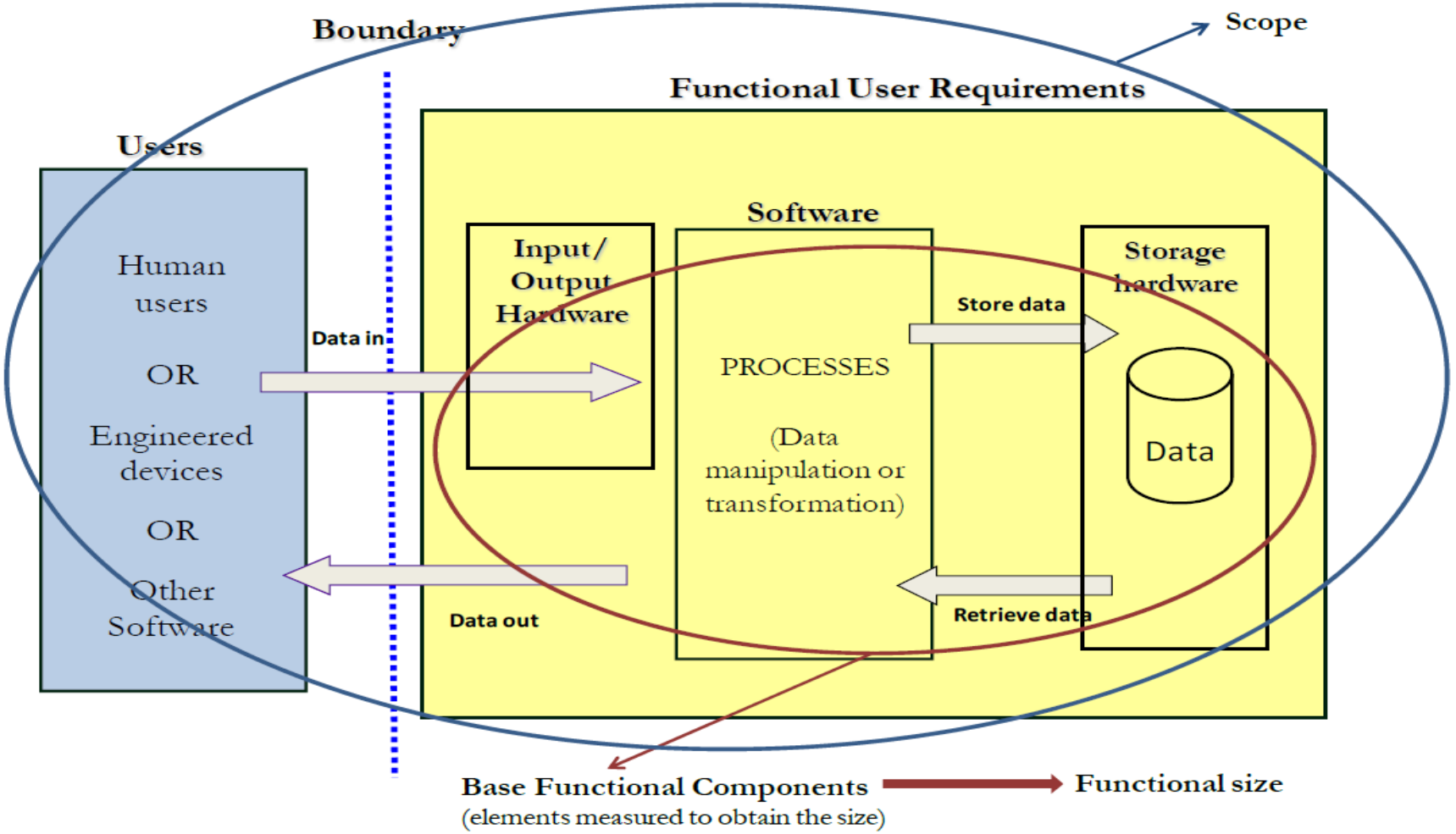
What is functional size measurement?

"Functional Size Measurement (FSM) is a technique used to measure the size of software by quantifying the Functional User Requirements of the software"

The current functional size measurement (FSM) methods:

- **ISO 19761:2011 COSMIC functional size measurement method**
- ISO 20926:2009 IFPUG 4.1 functional size measurement method
- ISO 24570:2005 NESMA functional size measurement method
- ISO 20968:2002 MKII function point analysis
- ISO 29881:2010 FiSMA 1.1 functional size measurement method

Functional size measurement methods: general representation [ISO 14143-1:2007]

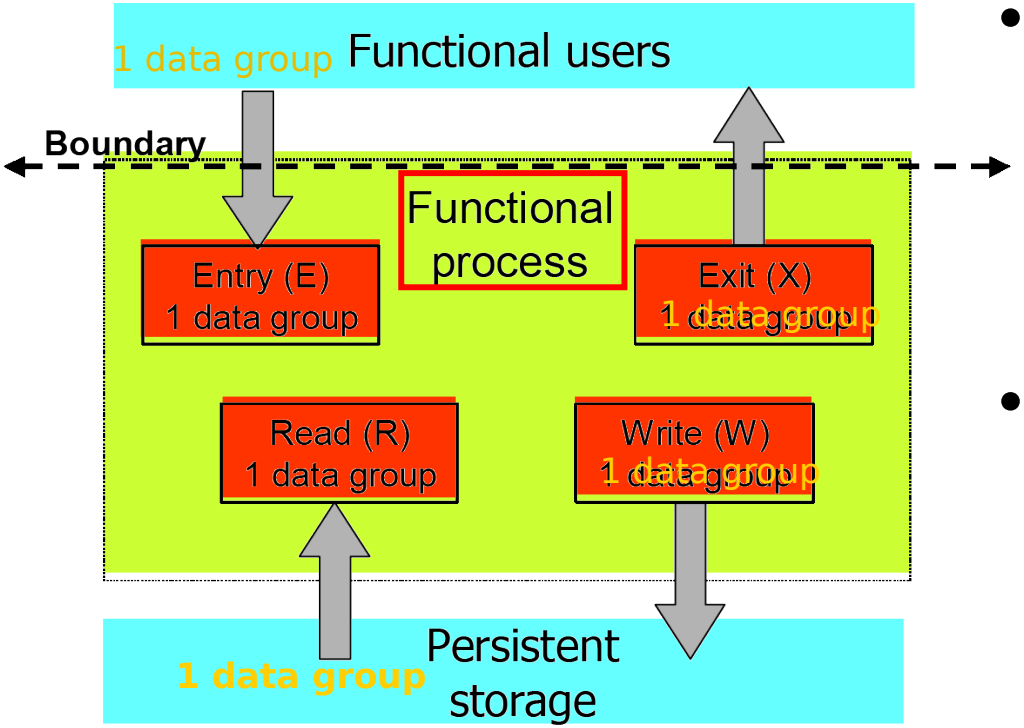


The COSMIC method

http://www.cosmicon.com/

1. The *Basic Functional Component (BFC) is the data movement.*
2. The measurement *unit is a COSMIC Function Point (CFP), which represents one data movement.*
3. A *functional user of a software application can be: a human, another software application, or a hardware device.*
4. A *boundary is a conceptual interface between the functional users and the software application.*
5. The functional users interact with the software application via *data movements.*
6. There are *four types of data movements: Entry (E), eXit (X), Read (R), and Write (W).*

COSMIC FSM Standard (ISO/IEC 19761)



- **System is viewed as a black-box**
- ***Size* =**

The number of movements of "***data-groups***" between the System and the Users / the Persistent Storage

- **Data-movements of 4 types:**
  1. ***Entry***
  2. ***Exit***
  3. ***Read***
  4. ***Write***
- **Unit of Size is**

Aggregate the Measurement Results

FP Size = Sum(Ne) + Sum(Nx) + Sum(Nr) + Sum(Mw)

Where:

- **Ne** = number of Entries
- **Nx** = number of Exits
- **Nr** = number of Reads
- **Mw** = number of Writes

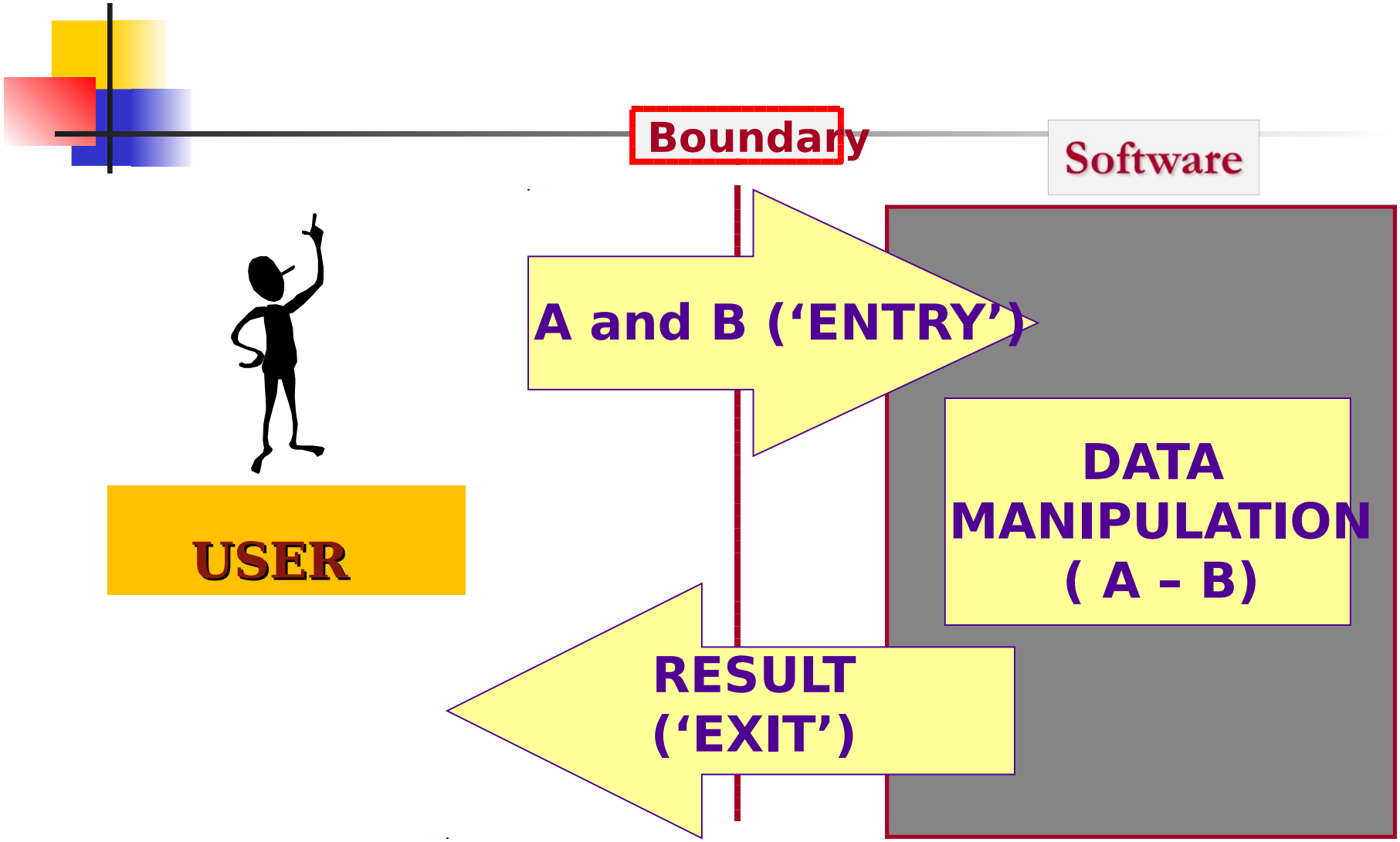
- **there is no upper limit to the functional size of the functional requirements**
- **Min size: 2 CFP**

Very Simple Exercise (1)

Apply COSMIC measurement procedure to measure "**A – B**":

- The user enters the values A and B of the operands
- The system in return displays the result

COSMIC for A-B



Exercise 1 Solution

No	Process ID	Process Description	Trigger	Sub-Process	Data Group	Sub-Process Type	FFP	
1	1.1	Subtraction of two operands	• Press Enter	- Read operands - Return result	Operands Result	E X	1 1	2

Total = 2 CFP

Functional process

A functional process is an elementary component of a set of Functional User Requirements, comprising a unique, cohesive and independently executable set of data movements.

functional processes & data movements

- Each functional process:
  - is independently executable
  - triggered by an event
  - Contains all that is required to be done in response to the triggering event
- Data movement types: E, X, R, W
- Entry type : considered to include certain associated data manipulations such as validation of the entered data
  - Triggering events are Entry type
  - Clock and timing events are Triggering events
- Each data movement moves data belonging to a single data group
  - data movement “de-duplication”

Measurement Units

DEFINITION – COSMIC unit of measurement
1 CFP (Cosmic Function Point), which is the size of one data movement.

PRINCIPLE – The COSMIC measurement principle
a) The size of a functional process is equal to the number of its data movements.
b) The functional size of a piece of software of defined scope is equal to the sum of the sizes of its functional processes.

- **Each data movement**
  - Is Counted only once within a functional process
  - Does not involve another data movement type

COSMIC measurement procedure

1. Identify the functional users
2. Identify the triggering events
3. Identify the functional processes enabled by those events
4. Identify the data groups
5. Identify the data movements from the interface (Entry, eXit, Read and Write)
6. Obtain the total number of Cosmic Function Points (CFP)

More Complex Exercise 2

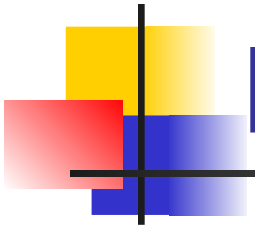
Scope of measurement: Measure the size of the functionality "Create a new customer" as it is described in the flow of events.

Functional user: Salesman

Pre-conditions: The salesman is already logged in the system and has selected the option "Create Customer"

Flow of events:

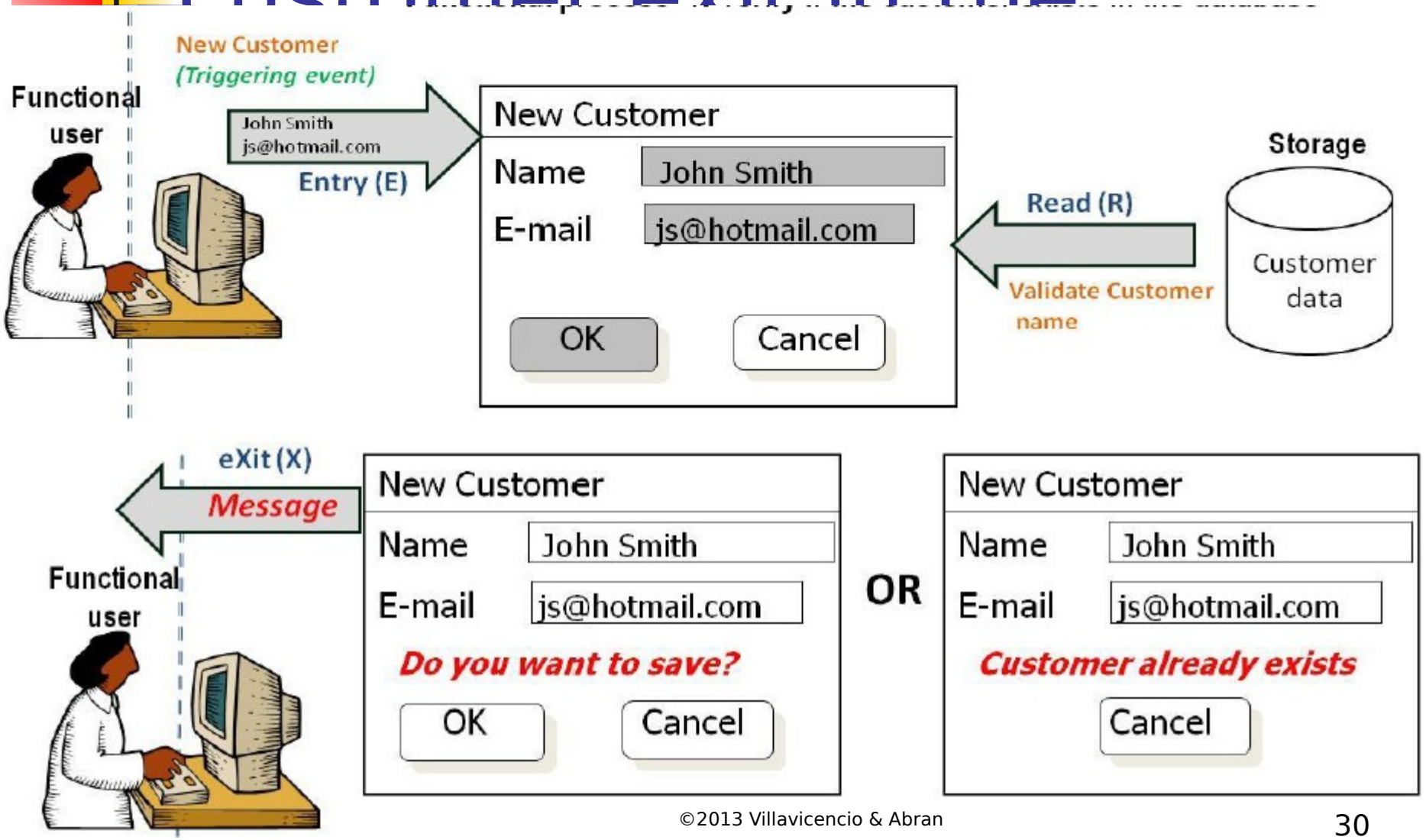
1. The salesman enters the name and email of the new customer (John Smith, js@hotmail.com) and press OK.
2. The system verifies if the customer already exists in the database.
3. If the customer exists, an error message is displayed.
4. If the customer is new, the system asks the salesman to confirm the data that is going to be saved by pressing OK. If needed, the salesman can correct the customer data.
5. A new customer is created into the database.
6. A confirmation or error message is displayed.



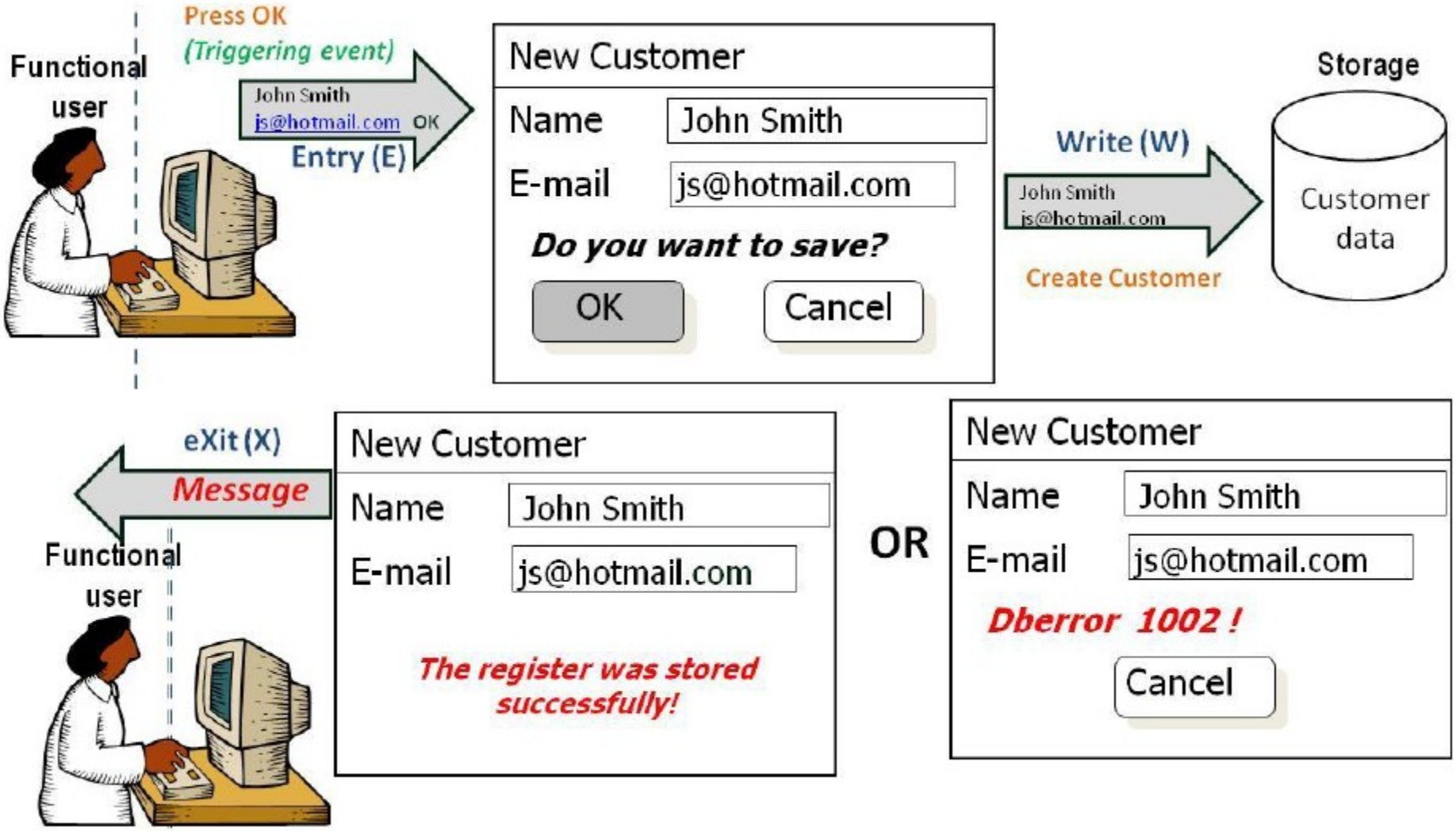
Exercise 2

- **Functional processes:**
  1) Verify if the customer exists in the database,
  2) Create a new customer in the database**.**
- **Data group (DG):**
  - Customer
  - System messages
- **Data attributes:**
  - Name, email

Exercise 2: Verify if the customer exit in the



Exercise 2: confirm customer data and save



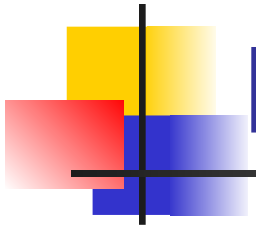
Exercise 3:
apply the COSMIC measurement procedure (in class, form teams of 2: 5

“Create a list of sales territories"

1 -This use case starts when the user requests to create a list of sales territories and enters the sales territories.

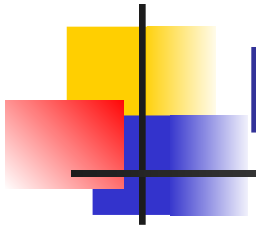
2 - The system saves the sales territories.

3 - The system displays a confirmation message or, if it fails to save the territories, it issues an error message.



Exercise 3: Solution

- In class



Exercise 4

- In class

Rule - data movement de-duplication

A data movement of a given type (E, X, R, W) moving a Data Group is *in general* identified only once in the Functional Process

- **Example1:** the developer decides to implement a Read of a data group by two commands to retrieve different subsets of data attributes of the same Object of interest from persistent storage at different points in the Functional Process.
- For sizing purposes, we identify only one Read

- **Example2:** suppose a Read is required in the FUR that in practice requires many retrieval occurrences, as in a search through a file.
- For sizing purposes, we identify only one Read

Rule - data movement de-duplication (identify data movements, not occurrences)

A data movement of a given type (E, X, R, W) moving any Data Group is *in general* identified only once in the Functional Process

- **Exception: if two Read of the same data type require different data manipulation, then the Read should be identified twice in the one Functional Process**

- **Example:** Repeat a Read before the process ceases in order to check that the data have not changed since the first Read in the same process.

Principles of the COSMIC FSM (1)

Each functional process consists of sub-processes.

A sub-process may be either a **data movement** or a **data manipulation.**

A data movement moves a single **data group** .

There are four data movement types, **Entry**, **Exit**, **Write** and **Read**. An Entry moves a data group into a functional process from a functional user. An Exit moves a data group out of a functional process to a functional user. A Write moves a data group from a functional process to persistent storage. A Read moves a data group from persistent storage to a functional process.

A data group consists of a unique set of **data attributes** that describe a single **object of interest.**

Each functional process is started by its **triggering Entry** data movement. The data group moved by the triggering Entry is generated by a functional user in response to a **triggering event.**

Principles of the COSMIC FSM (2)

a) Software is bounded by hardware.

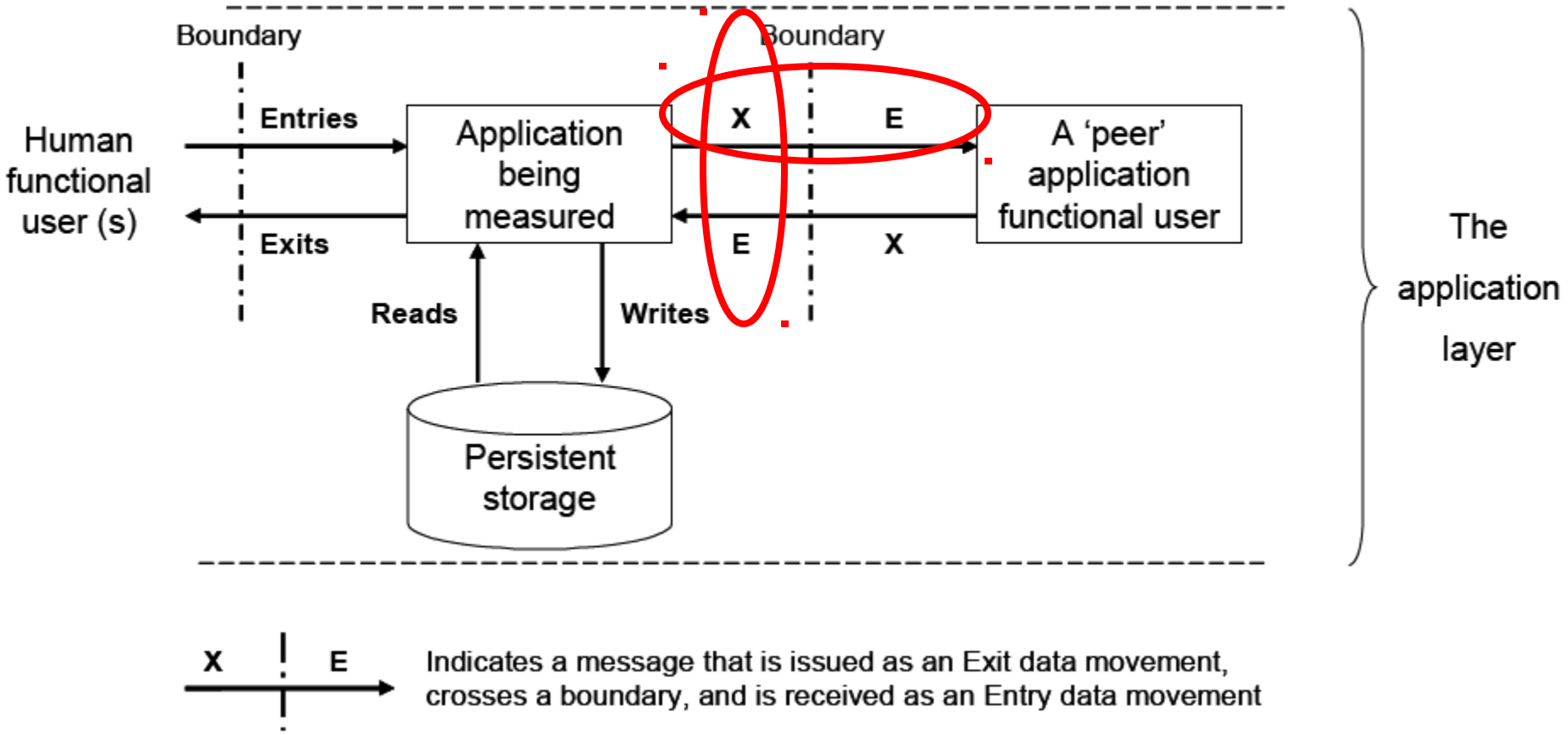
b) Software is typically structured into **layers.**

c) A layer may contain one or more separate '**peer' pieces of software.**

d) Any piece of software to be measured, shall be defined by its measurement **scope, which shall be confined wholly within a single layer.**

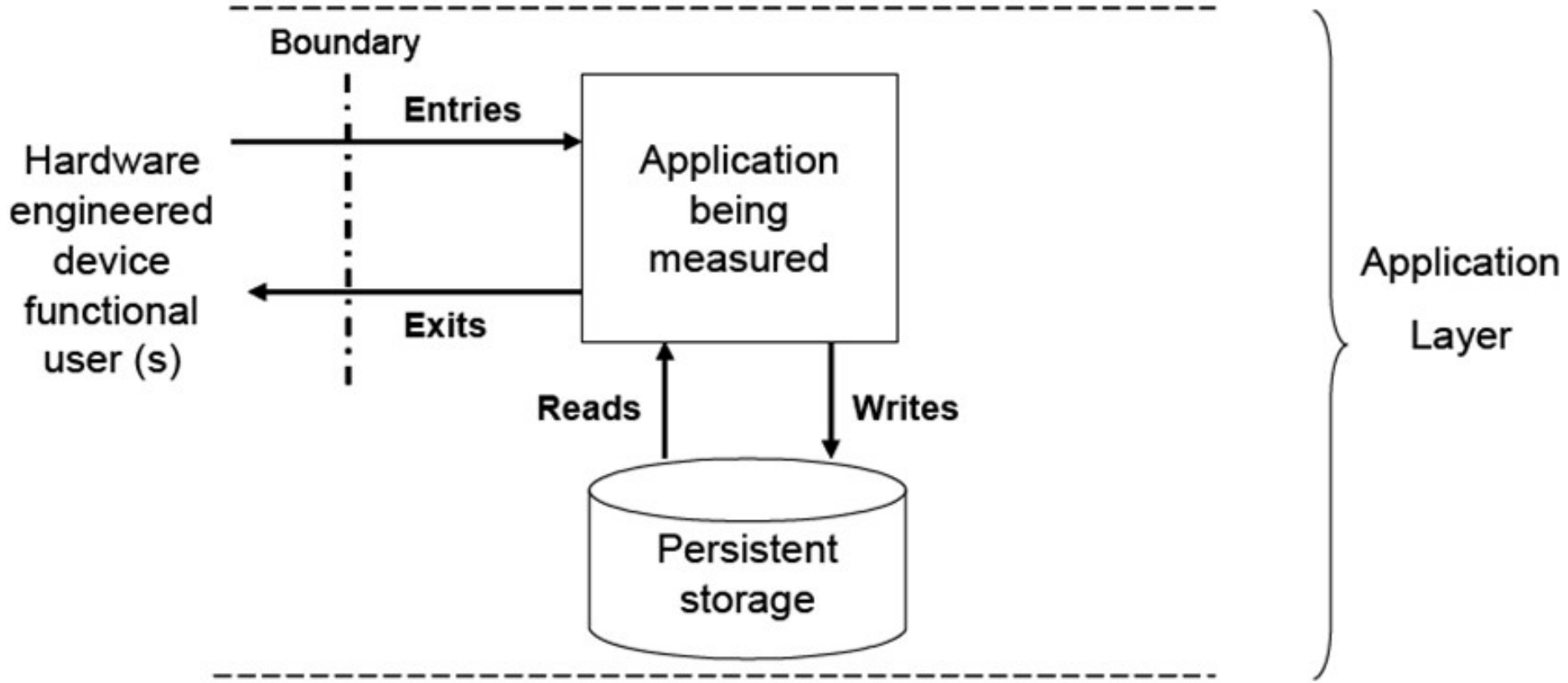
COSMIC: Logical View on Business Application

(all previous examples belong to this category)

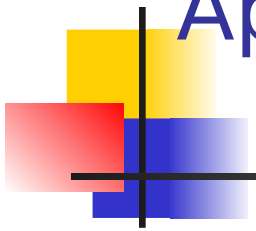


COSMIC: Logical View on Real-time Embedded Application

(next example)



Applicability of COSMIC functional size measurement method



- Business application software
- Real-time software
- Hybrids of the above

Calculation of Functional Size of Changes to the FUR

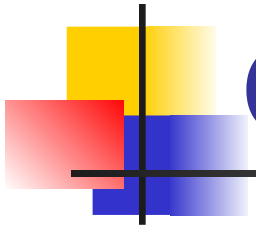
$$Size_{Cfsu}(Change) = Sum_{size}(added\ data\ movement_i) + Sum_{size}(changed\ data\ movement_i) + Sum_{size}(deleted\ data\ movement_i)$$

- **Definition for change (ISO Standard):**
  - a data movement is considered to be changed if any of the attributes of the data group are changed, or of any changes are needed to the data manipulat associated with the data movement

Calculation of Functional Size of Changes to the FUR

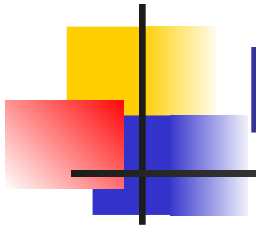
$$Size_{Cfsu}(Change) = Sum_{size}(added\ data\ movement_i) + Sum_{size}(changed\ data\ movement_i) + Sum_{size}(deleted\ data\ movement_i)$$

- **Definition for change (ISO Standard):**
  - a data movement is considered to be changed if any of the attributes of the data group are changed, or of any changes are needed to the data manipulat associated with the data movement



Questions?

- ...



Next?

- Review for the exam
- Exam on October 17,
  - in class, closed book
  - Coverage: L1 to L12 (incl.)