Experimental Software Engineering: Role and Impact of Measurement Models on Empirical Processes

by
Giovanni Cantone
Experimental Informatics, ESE
Dept. of Informatics, Systems and Production Eng., DISP
University of Rome at Tor Vergata Campus, UNIRM2

cantone@uniroma2.it

Outline

- ESE (E^2SE)
- EPM (Typical)
- Case 1: Traditional & Simple
- IMPACT OF MM ON EPM
- Case 2: New & Complex

ESE & Modeling

1- Development and continual improvement of empirical-evidence-based software models.

2- Capitalization organization wide of the results.

Basic components

EPM

- GQM
- MMLC
- QIP
- EXPERIENCE FACTORY
- Applied Statistics

Empirical strategies

Survey
Case study
Experiment

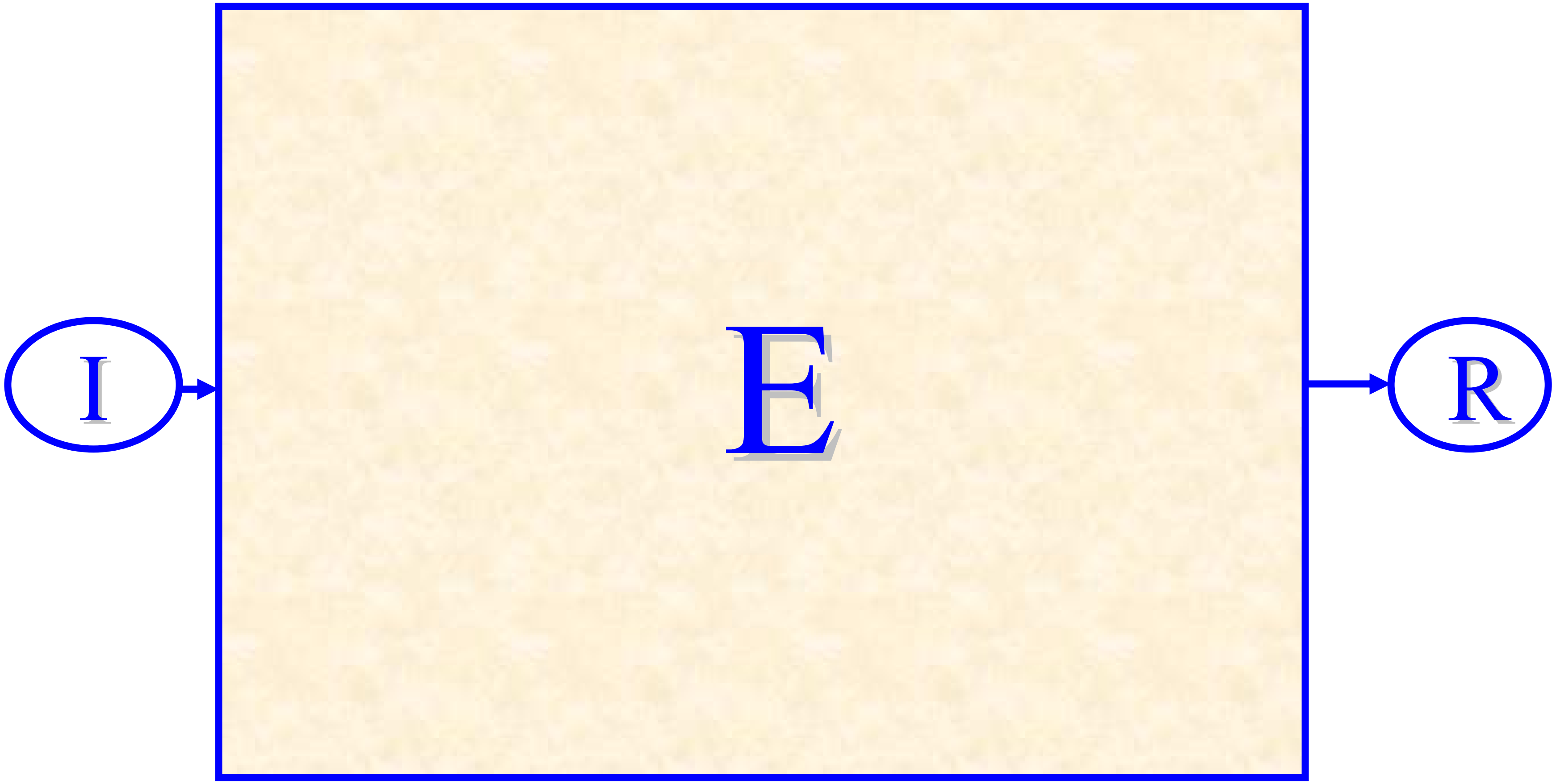
See for instance[C. Wohlin et others.: Experimentation in SE, Kluwer AP]

Types of experiment context

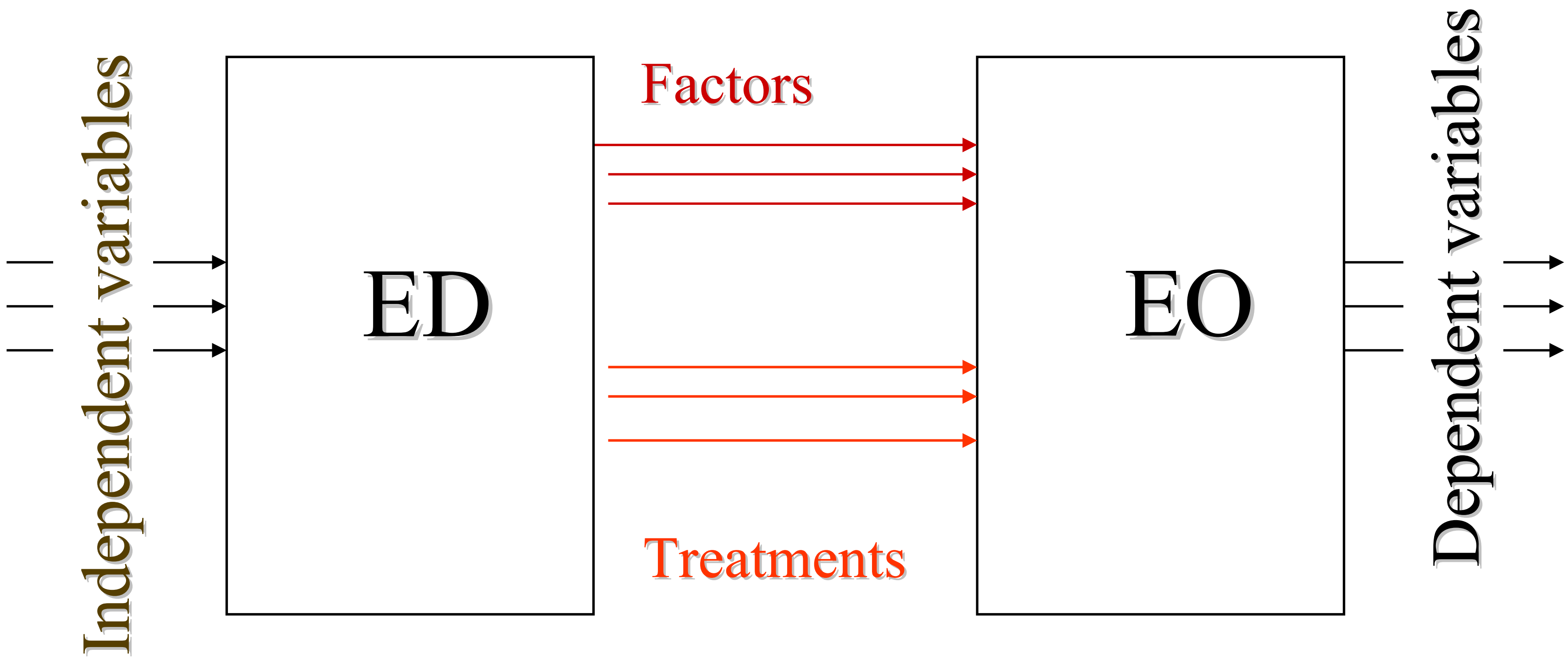
Team | Subjects / Objects	One	Many
One	Single object study	Variation study on multiple objects
Many	Single object multi-test study	Blocked subjects and objects study

Replication

The Experiment

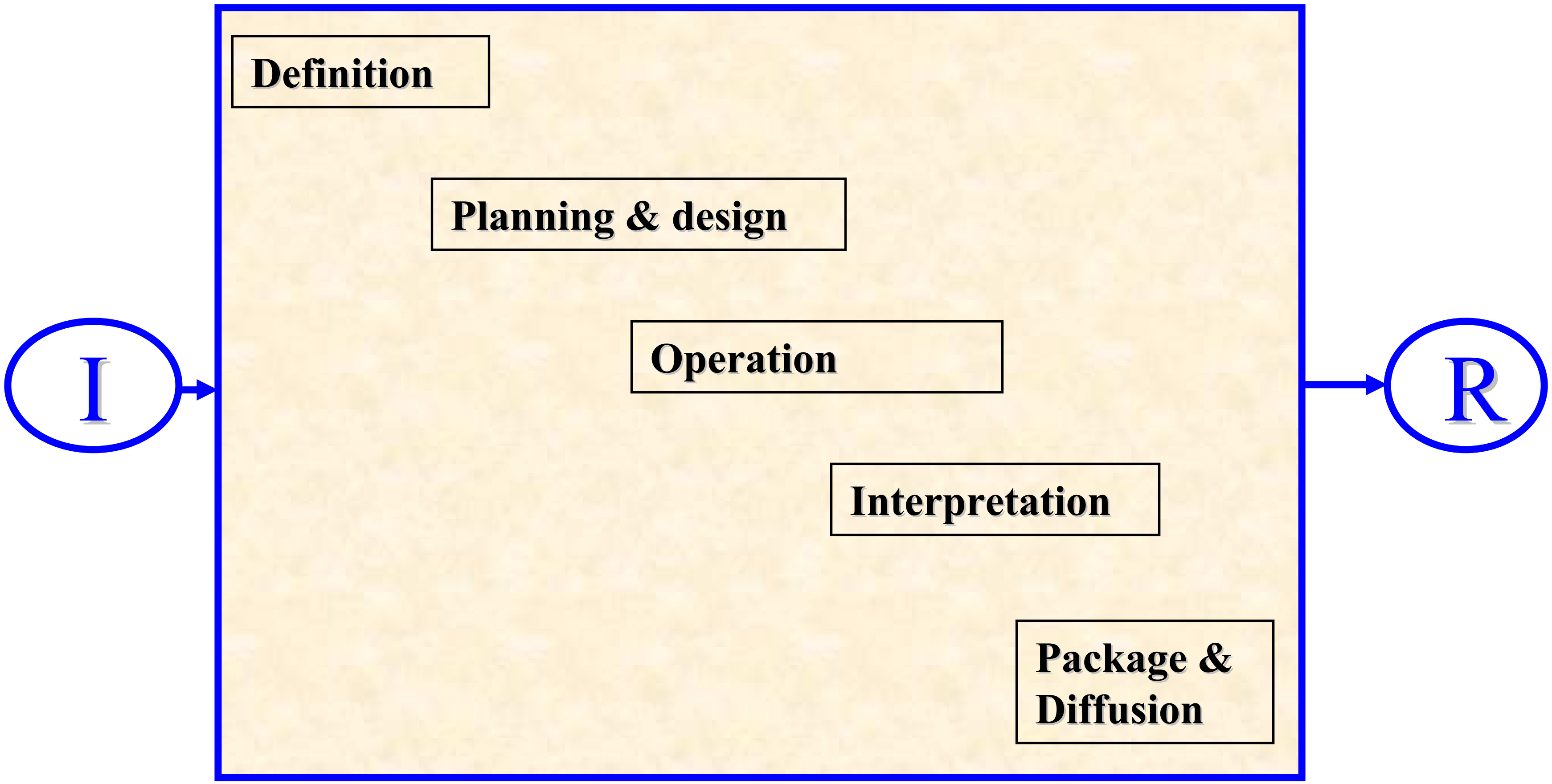


The Experiment

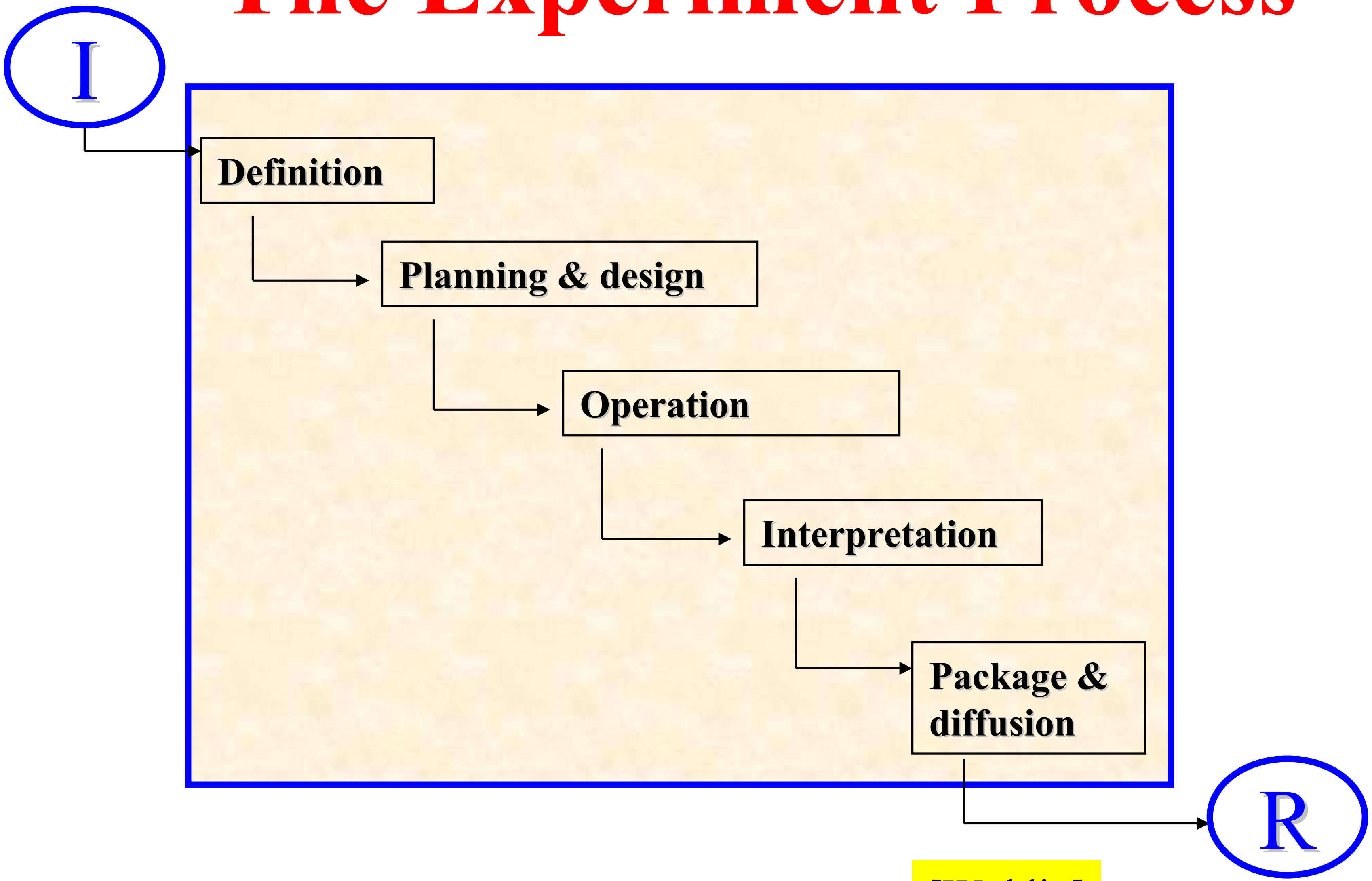


[Wohlin]

The Experiment

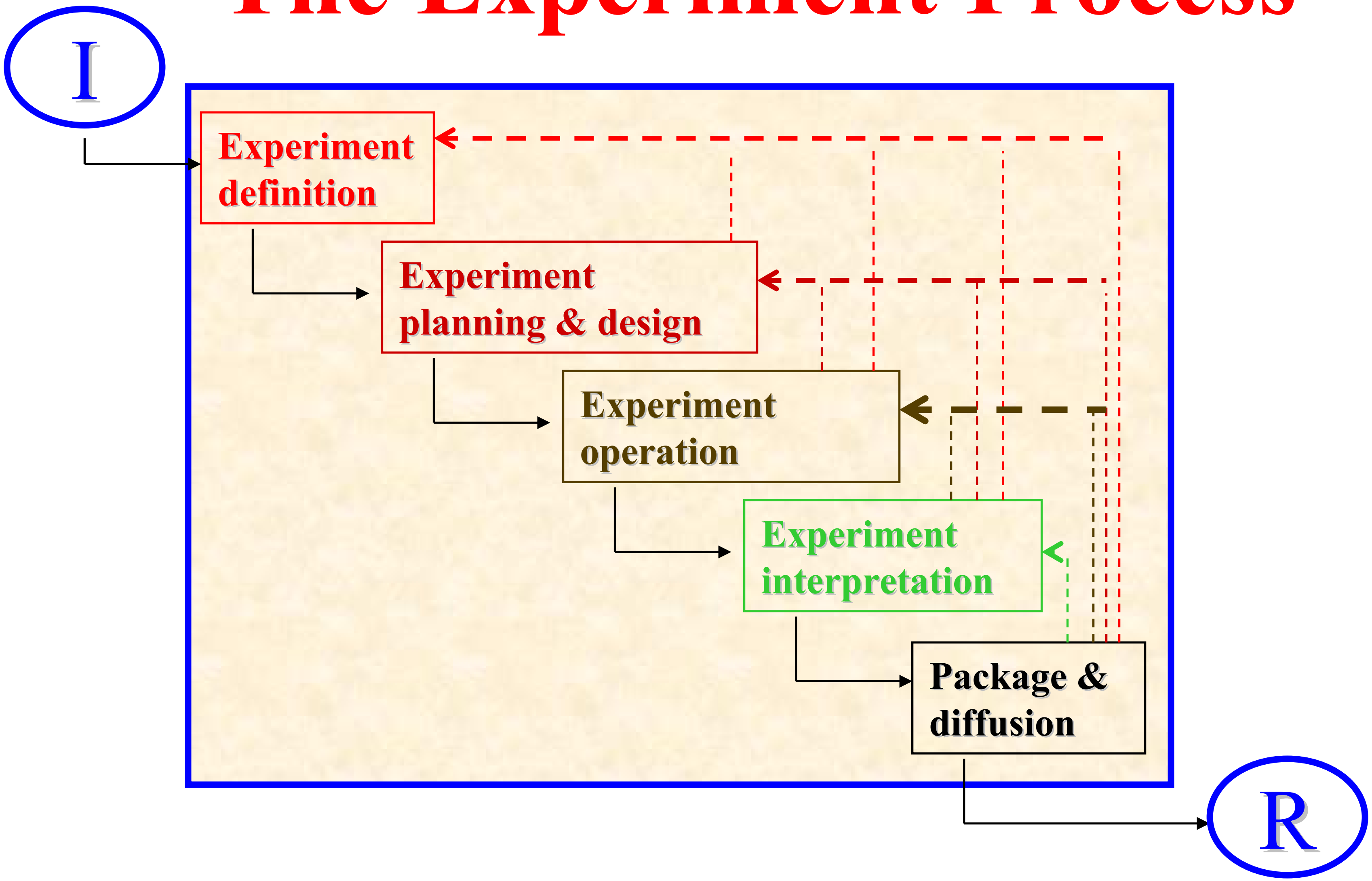


The Experiment Process

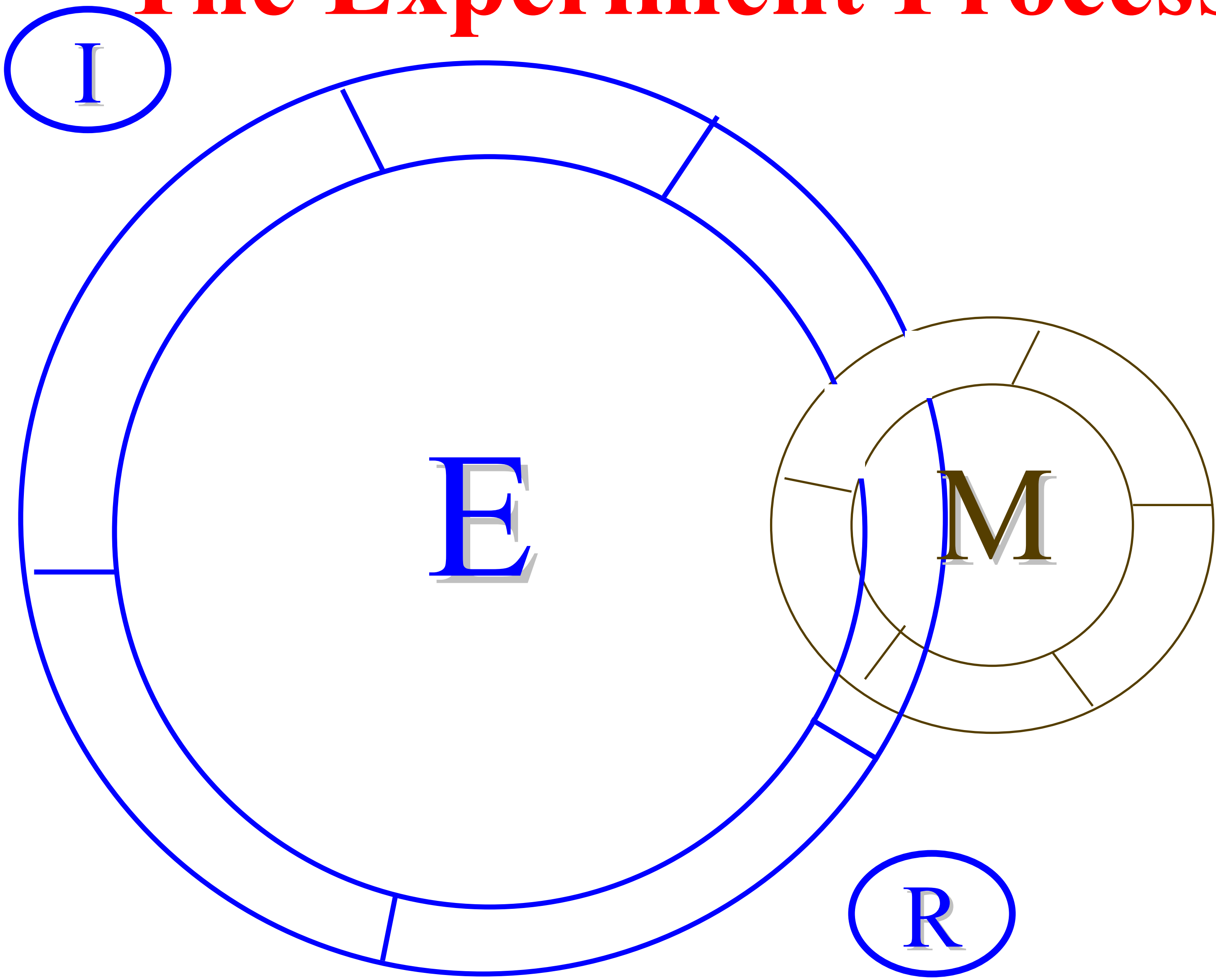


[Wohlin]

The Experiment Process



The Experiment Process



CASE I

SW Quality : Benefits and Costs of CCM

↓

Inspection of HL CCM SW: Effectiveness and Efficacy

CASE I

SW Quality : Benefits and Costs of CCM Inspection of CCM SW

Inspection of Analysis and Design SW CCM Documents: Effectiveness and Efficacy

CCM SW Documents

!CCM SW Documents

CASE I

SW Quality : Benefits and Costs of CCM Inspection of CCM SW

Defects of Inspected SW Entering Testing

CCM
SW

!CCM
SW

CASE I

SW Quality : Benefits and Costs of CCM Inspection of CCM SW

Finding Seeded Defects: Efficiency & Efficacy

Code reading

Testing

Experiment Process

Definition

Object	**Strategies of *Code-reading* and of *Functional testing*.**
Purpose	**To evaluate performance**
Quality focus	**Efficiency & Efficacy in finding seeded defects.**
Perspective	**End-user.**
Context	**URM2.DISP.ESEG + X.Z.com**

Measurements

Counting measures

- Number of faults found
- Duration of the session
- Fault Classification

Simple Indirect measurements

Experiment Process

Experiment planning

- Context selection
- Hypothesis formulation
- Variable selection
  - Independent variables
    - Treatments (Code reading, F. Testing)
    - Factors: (Categorized faults)
  - Dependent Variables

Experimental Process

Fault Categories

- **Initialization**: e.g. wrong initialization of attributes
- **Computing**: e.g. wrong computations of variables
- **Control**: e.g. wrong definitions of logic variables.
- **Building and using complex structures of data**: e.g. inserting elements unrelated to the logic and the structure of data.
- **Graphical interface**: e.g. wrong settings of interface windows.
- **Functionality**: e.g. wrong realization of functionality.
- **Events managing**: e.g. wrong management of event.
- **Exceptions handling**: e.g. unforeseen produced exception

Experimental Process

- Experiment design
- Instrumentation
  - Objects
  - Guidelines
  - Measurement instruments
- Validity evaluation

Experimental Process

- Subject training
- Context selection
  - Mode of work
  - Types of subjects
  - Type of application
  - Type of validity

Experimental Process

Operation

- Preparation
  - Execution

Experimental Process

Pre-Analysis & Interpretation

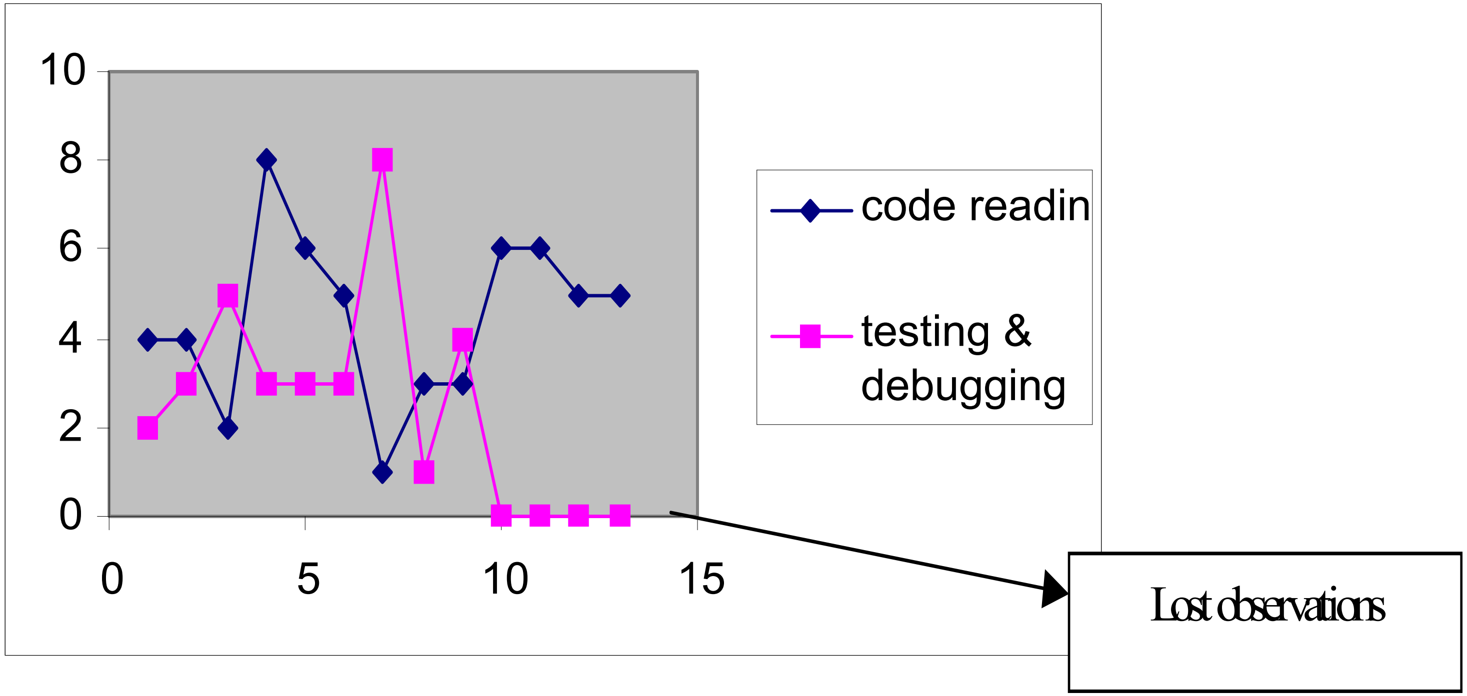
•Data collected by subject groups

Code reading			Testing & debugging		
Group numbers	Number of faults detected	Average time related Per minute	Groups numbers	Number of faults detected	Average time related Per minute
1	4	86	1	2	97
2	4	111	2	3	60
3	2	118	3	5	37
4	8	49	4	3	90
5	6	115	5	3	94
6	5	111	6	3	68
7	1	115	7	8	108
8	3	108	8	1	81
9	3	94	9	4	97
10	6	83	10	0	0
11	6	68	11	0	0
12	5	42	12	0	0
13	5	102	13	0	0

• Results

Code reading		Testing & debugging	
Average of total faults detected	Average time related Per minute	Average of total faults detected	Average time related Per minute
4	92 min	2	56 min

• Data Reduction



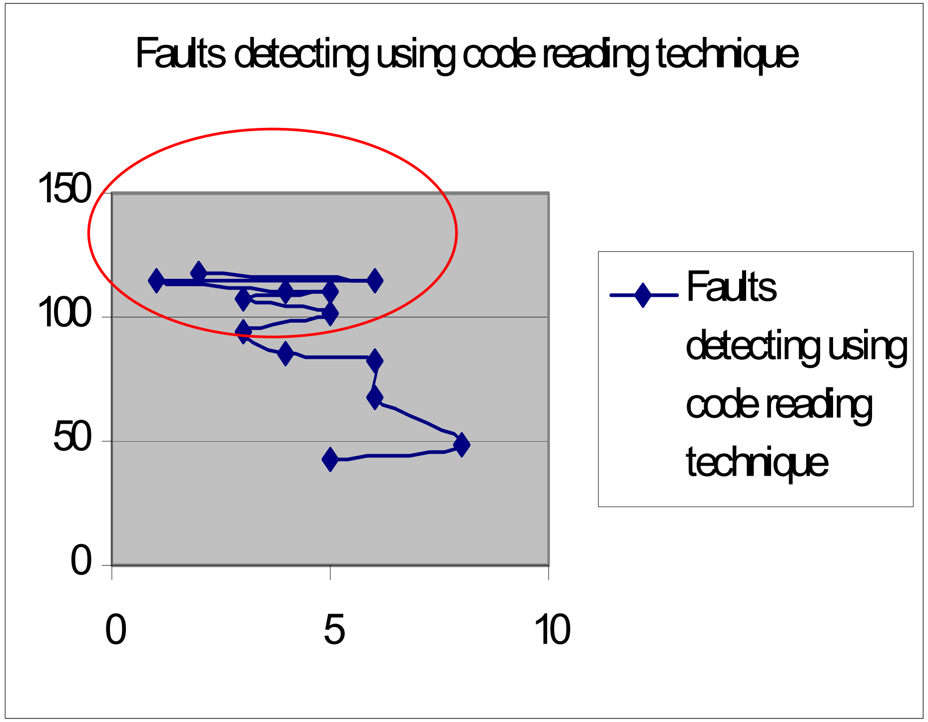
• Enhanced results

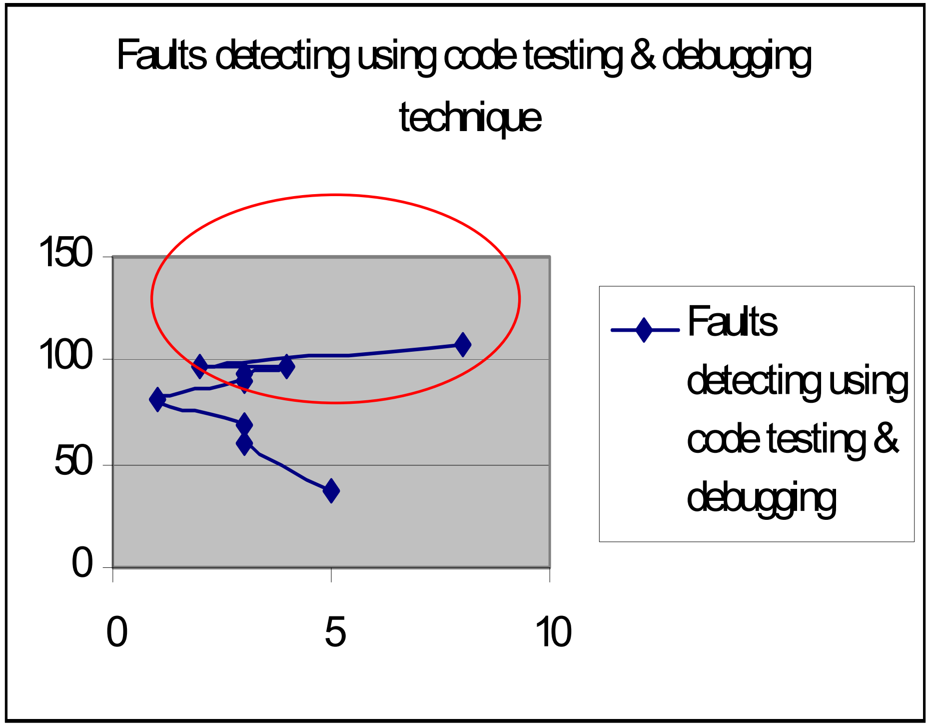
Code reading		Testing & debugging	
Average of total faults detected	Average time related Per minute	Average of total faults detected	Average time related Per minute
4	92 min	4	79 min

• Faults detection factor

Subjects	CR	T&D
Number of faults detected	58	32
Average number of faults detected	4	4
Average time for detected faults	92	79

• Classification of collected data





Data shown in circle are faults detected after 70-80 of the experiments impact

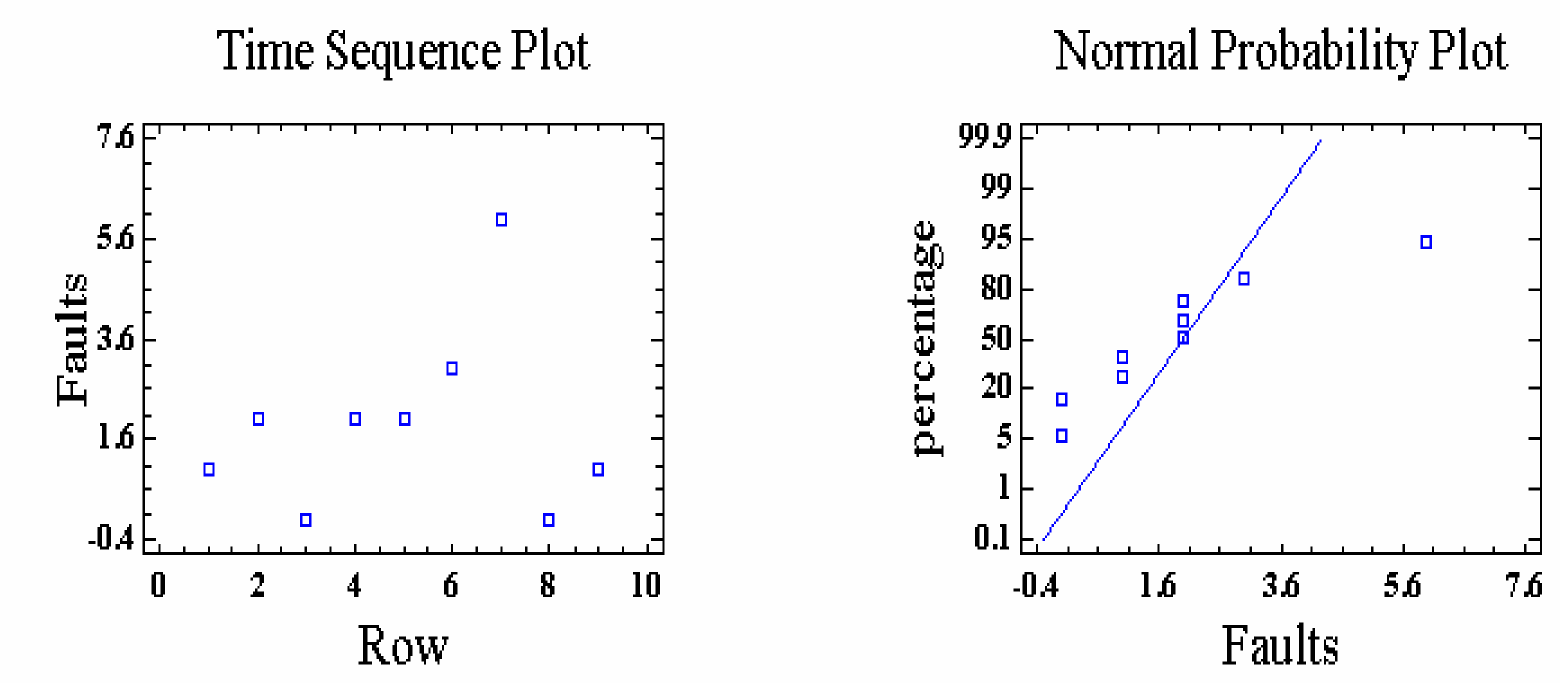
• Reclassification process of collected data

Classification type	Faults detected using Code reading	Percentage %	Percentage out of 38 seeded faults	Faults detected using Testing & debugging	Percentage %	Percentage out of 38 seeded faults
0	10	17	0.01	7	22	0.01
1	10	17	0.01	8	25	0.01
2	38	66	0.03	17	53	0.01
Total	58	100		32	100	

• Analysis of statistical approaches

• One sample analysis

• Distribution using testing and debugging and their mean

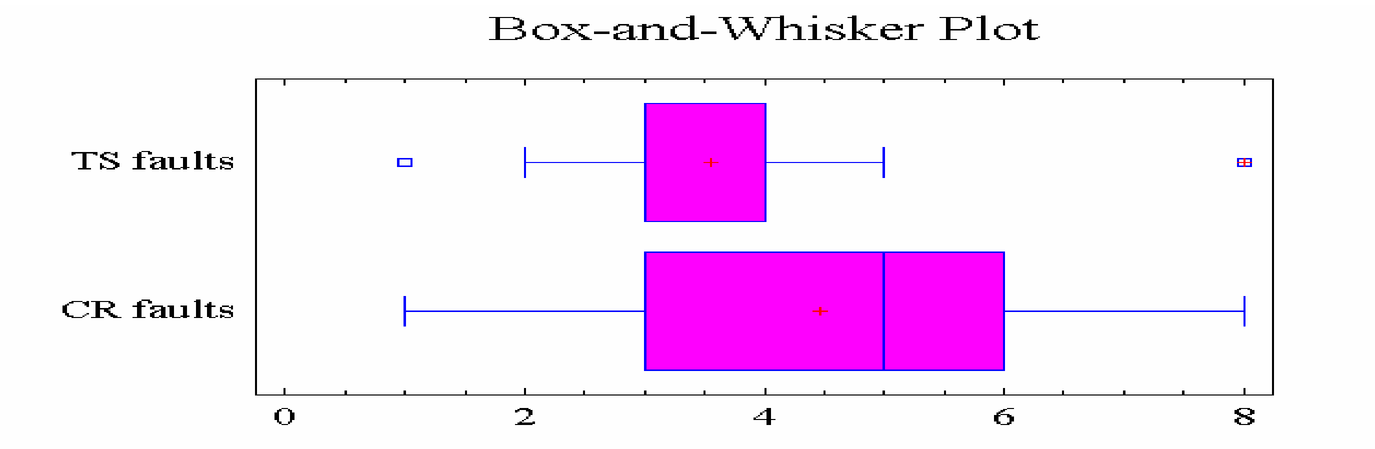


Type 2 Faults. T & D

• Analysis of statistical approaches

• One sample analysis

• Difference between CR and T & D techniques



Data distributions and mean

• Analysis of statistical approaches

• One sample analysis

• t-Test: Two-Sample Assuming Equal Variances

	Testing & debugging (TS)	Code reading (CR)
Mean	2.428571429	3.454545455
Variance	2.952380952	3.472727273
Observations	7	11
T calculated	-1.172108115	
t Critical two-tail	2.119904821	
p(T<=t) one-tail	0.129155674	

t-test for the hypothesis for Type2 faults only

• Analysis of statistical approaches

• One sample analysis: Code reading

Total fault	4	4	2	8	6	5	1	3	3	6	6	5	5
Average time	86	111	118	49	115	111	115	108	94	83	68	42	102

• The equations of regression

x on y Faults = 8.63742 - 0.0451634*Time

y on x Time = 129.075 - 8.20641*Fault

P-values (ANalysis Of VAriance table) < 0.05 => Statistically significant relationship between Fault and Time at the 95% confidence level.
R Square statistics: Variability in Fault 37.0629% ; Correlation coefficient = -0.608793 => Moderately strong relationship

• Analysis of statistical approaches

• One sample analysis: Testing & Debugging

Total faults	2	3	5	3	3	3	8	1	4
Average time	97	60	37	90	94	68	108	81	97

• The equations of regression

x on y Faults = 2.642 + 0.0112322*Time

y on x Time = 76.331 + 1.4069*Faults

P-values (ANOVA table) < 0.10 => Statistically ! significant relationship between Fault and Time at the 90% confidence level.
R-Squared statistic: Variability in Time =1.58026% ; Correlation coefficient = 0.125708 => Week relationship

CASE II
Technology Transfer: Evaluation of Competing Software Technologies

↓

What kind of technology?

↓

Workflow Automation, WA, development suites & engines

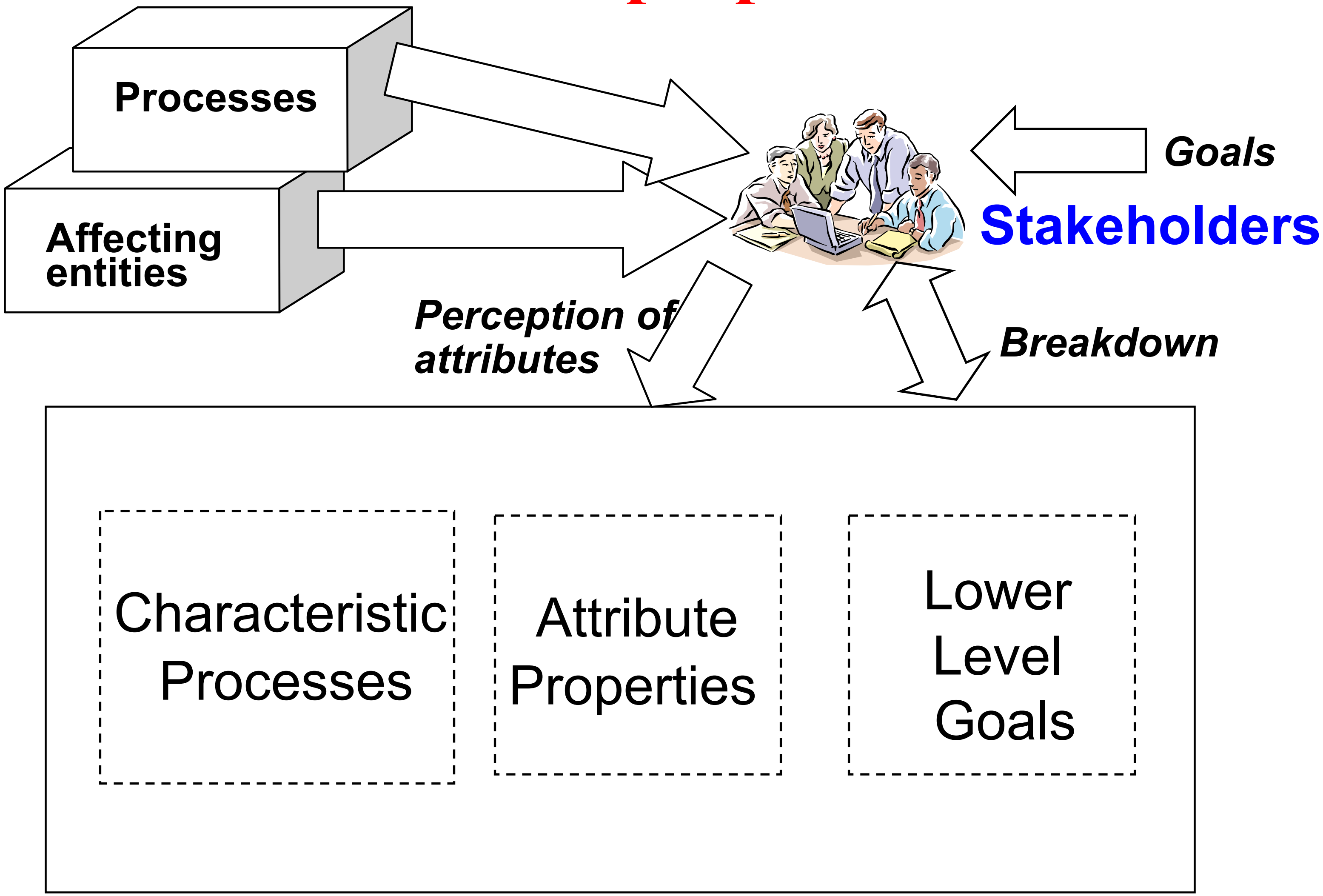
Goal

Object	**WA Technologies.**
Purpose	**To identify the "best" tech.**
Quality focus	**Organization HL goals.**
Perspective	**Research**
Context	**URM2.DISP.ESEG && Org/Dept/A.Italy.Admin**

Case study: Goal

Object	**WA Technologies.**
Purpose	**To identify the "best" tech. (for certain stated HL goals).**
Quality focus	**Goal-driven & Characteristic-processes-based comparison.**
Perspective	**Research**
Context	**URM2.DISP.ESEG && Org/Dept/A.Italy.Admin**

Goal breakdown, entities, their attributes and properties.



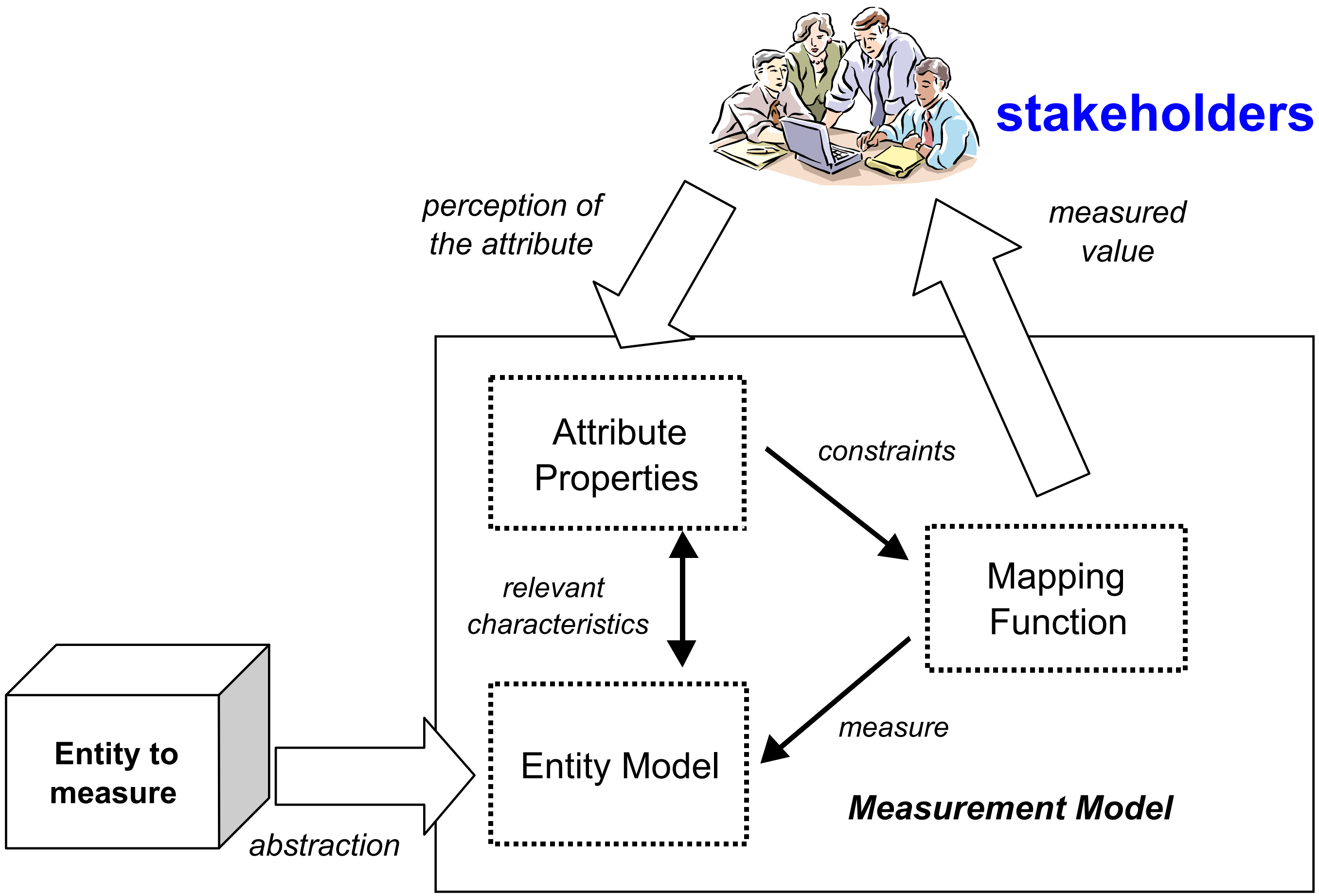
Case study: Goal

Object	**WA Technologies.**
Purpose	**To identify the "best" tech.**
Quality focus	**Adequacy to Reference organization Goals and Characteristic-processes.**
Perspective	**Research**
Context	**URM2.DISP.ESEG && Org/Dept/A.Italy.Admin**

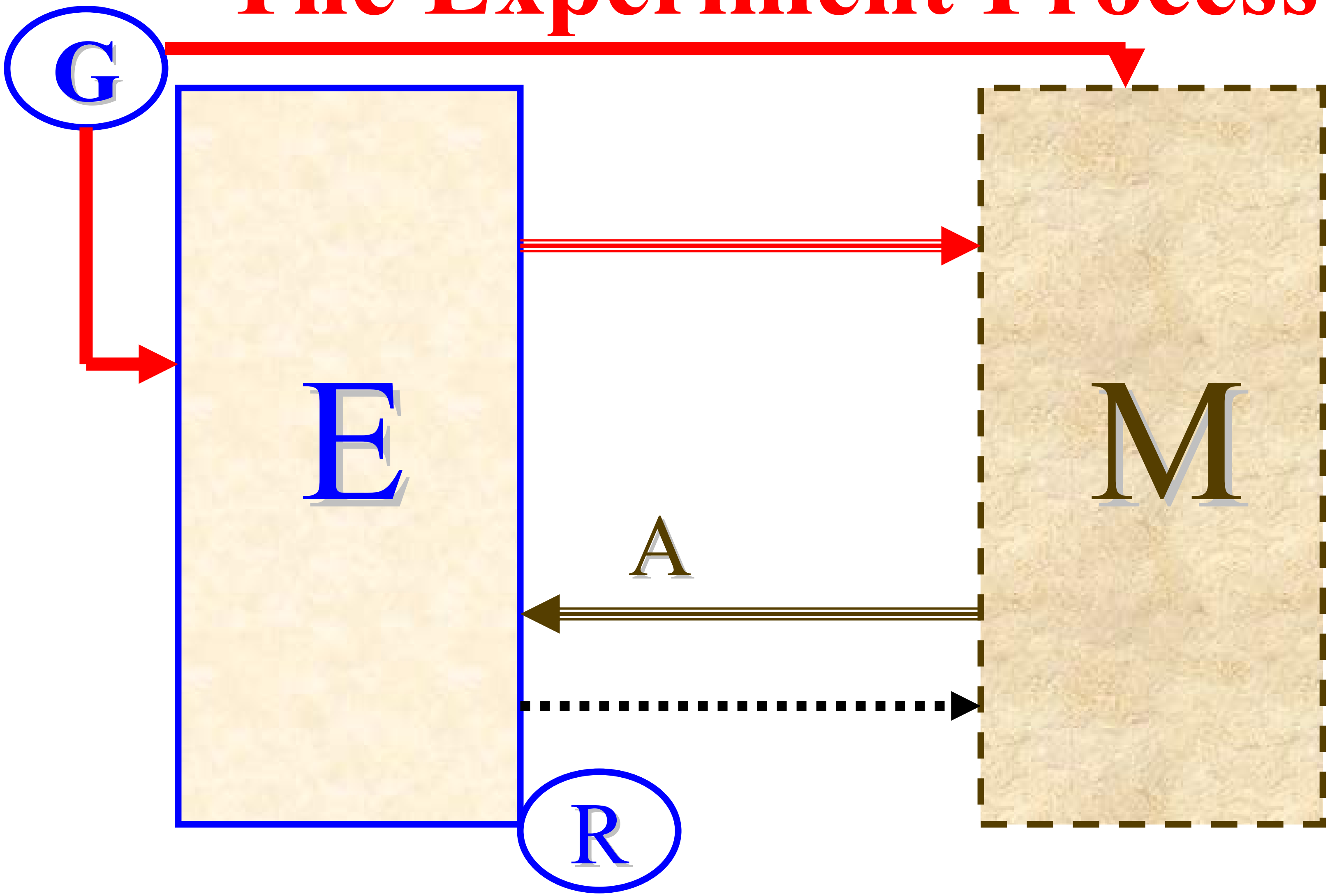
Goal breakdown, entities, their attributes and properties.

1- Goals$^+$
2- Questions$^+$
3- Metrics$^+$

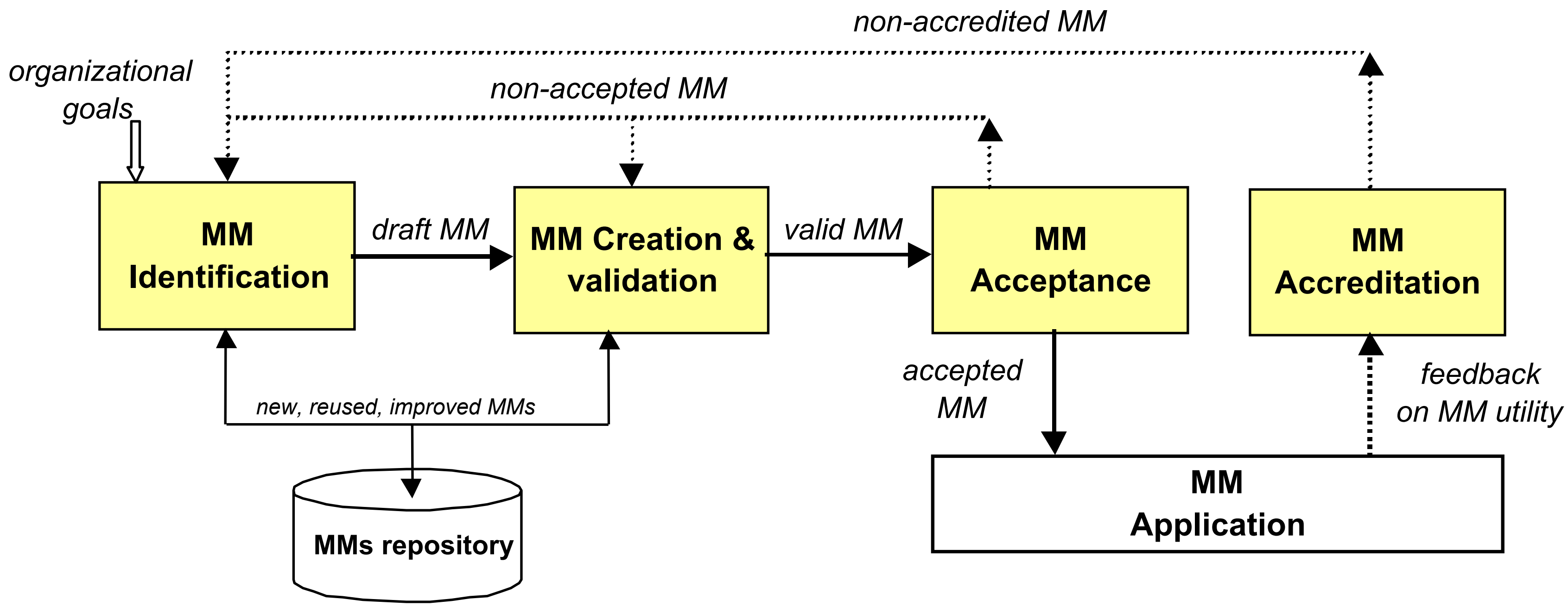
Stakeholders of a Measurement Model



The Experiment Process

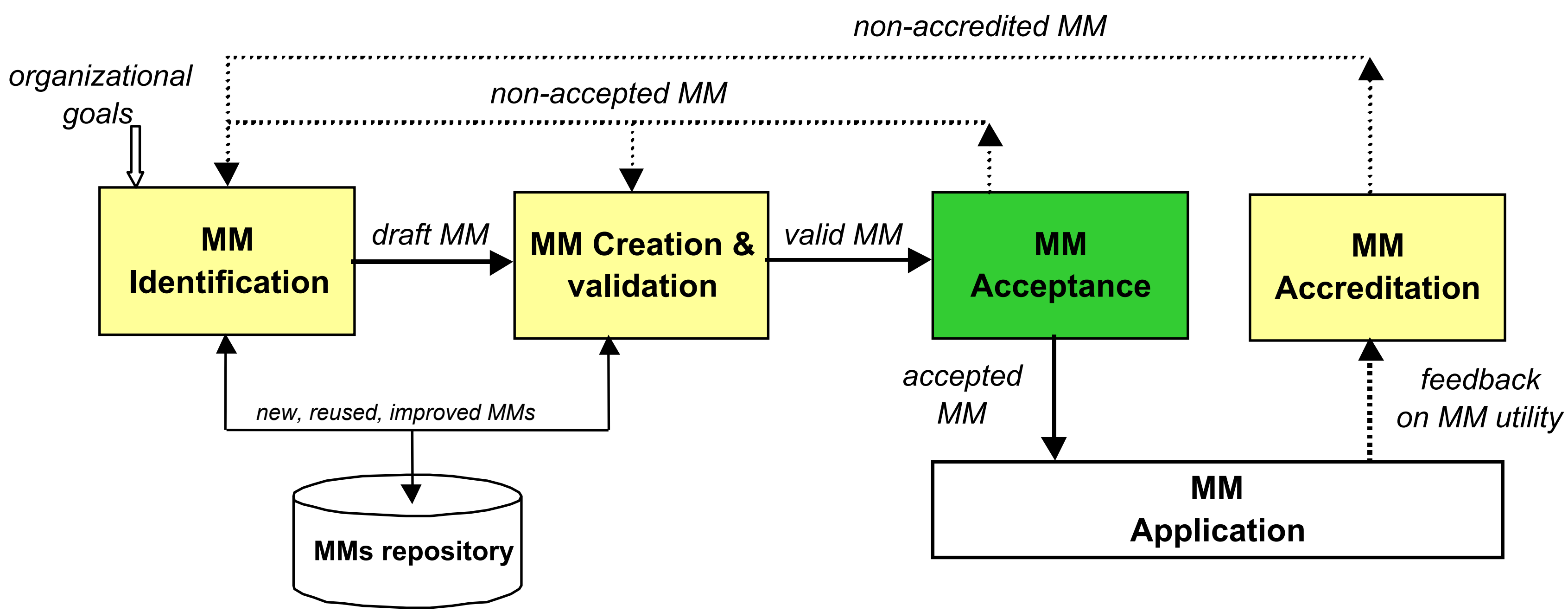


A Measurement Model Life Cycle for software



ESEG-UNIROMA2-MMLC

Acceptance is experiment too



ESEG-UNIROMA2-MMLC

From goals to measurement models

1. Identifying and characterizing quantitatively High Level Goals, HLG.

From goals to measurement models

1. Identifying and characterizing quantitatively High Level Goals, HLG.

2. Analyzing HLG, deriving and tracing Technical Analysis Goals, TAG.

From goals to measurement models

1. Identifying and characterizing quantitatively High Level Goals, HLG.
2. Analyzing HLG, deriving and tracing Technical Analysis Goals, TAG.
3. Deriving further Constrains, HLC, from the organization's vision.

From goals to measurement models

1. Identifying and characterizing quantitatively High Level Goals, HLG.
2. Analyzing HLG, deriving and tracing Technical Analysis Goals, TAG.
3. Deriving further Constrains, HLC, from the organization's vision.
4. Deriving Design Goals, DTG, from HLC, HLG e ATG.

From goals to measurement models

2. Analyzing HLG, deriving and tracing Technical Analysis Goals, TAG.
3. Deriving further Constrains, HLC, from the organization's vision.
4. Deriving Design Technology Goals, DTG, from HLC, HLG e ATG.
5. Giving DTG a structure.

From goals to measurement models

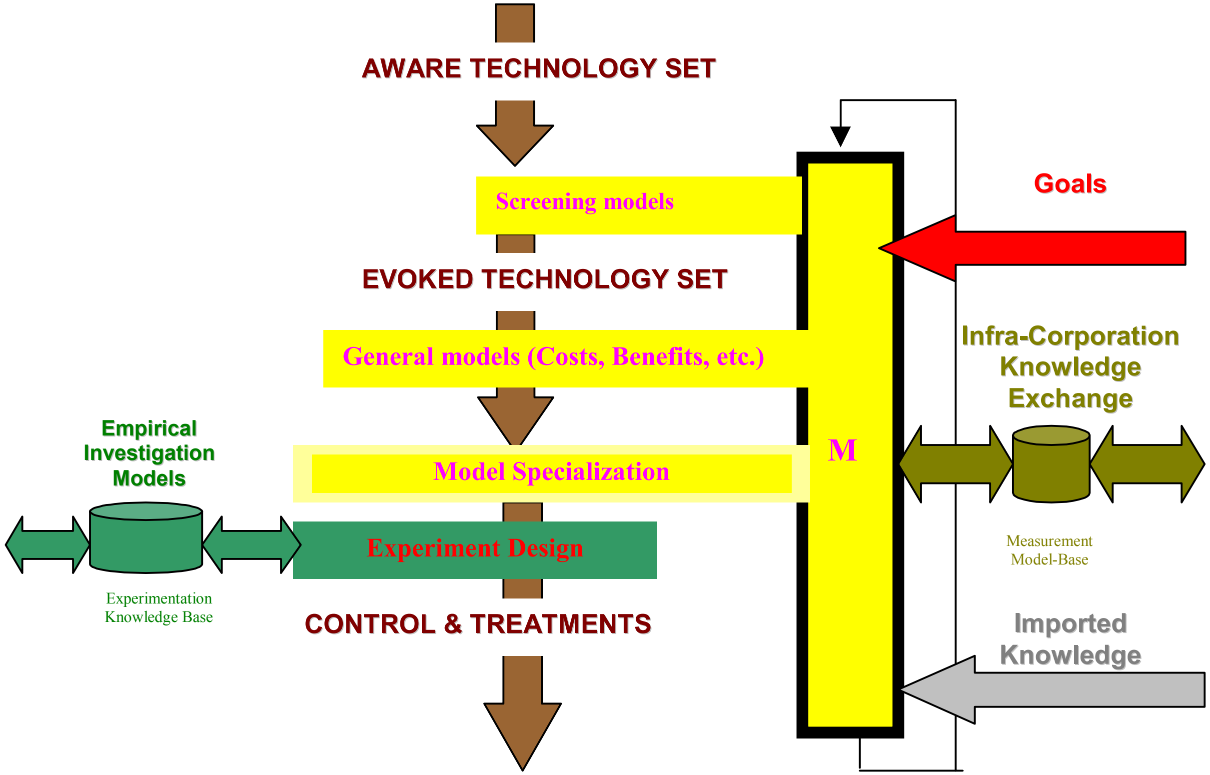
3. Deriving further Constrains, HLC, from the organization's vision.
4. Deriving Design Technology Goals, DTG, from HLC, HLG e ATG.
5. Giving DTG a structure.
6. Involving *stakeholders*, deriving metric attribute from SDTG and weighting such attributes level after level.

From goals to measurement models

5. Giving DTG a structure.
6. Involving *stakeholders* and weighting DTG.
7. Developing measurement models, TTMM, to filter, evaluate, and eventually select the control & treatment technologies.

From goals to measurement models

Using measurement models, TTMM, to filter, evaluate, and eventually select control & treatment technologies.



From goals to measurement models

5. Giving DTG a structure.
6. Involving *stakeholders* and weighting DTG.
7. [Developing, and] using measurement models, TTMM, to filter and evaluate the technology *Awareness set*.
8. Verifying and Accepting TTMMs && Redesigning original TT case study or experiment in order to reuse the incoming new empirical evidence.

A Process Model for Experimenting with WAT URM2-MSS

- **M**ultiple developments of EUO's Synthetic Processes
- **S**ingle development of a "Laboratory Project", i.e., a scaled–down realistic case study: less than a pilot project but much more than a toy project.
- **S**ingle development *of a Pilot Project that is an observed field project.*

NB: One more MM, EP (at least!)

Selected Data from an ESEG DISP URM2 Experiment

The TT Experiment Process: Design

Allocation of Groups for Parallel Development of Synthetic Processes

Phase	Step	Control Group	Activity	Treatment Group	Activity	Synthetic Process
0	0.1	A, B, C	0.1.1	A, B, C	0.1.2	
						Training
	0.2	A, B	0.2.1	A, B	0.2.2	SP_T
						Training
1	1.1	B	1.1.1	A	1.1.2	SP_1
2	2.1	A	2.1.1	B	2.1.2	SP_2
3	3.1	A	3.1.1	-	-	SP_3
		-	-	B	3.1.2	SP_4

The TT Experiment Process: Design

Allocation of a further group for Virtual Parallel Development of Synthetic Processes

Phase	Step	Control Group	Activity	Treatment Group	Activity	Synthetic Process
0	0.1	A, B, C	0.1.1	A, B, C	0.1.2	Common Training
	0.2	A, B	0.2.1	A, B	0.2.2	SP_T Training
		C	0.2.3	C	0.2.4	
1	1.1	B	1.1.1	A	1.1.2	SP_1
		C	1.1.3.n	C	1.1.4.n	SP_3
2	2.1	A	2.1.1	B	2.1.2	SP_2
		C	2.1.3.n	C	2.1.4.n	SP_4
3	3.1	A	3.1.1	-	-	SP_3
		-	-	B	3.1.2	SP_4
		C	3.1.3.n	C	3.1.4.n	SP_2
4	4.1	C	4.1.1.n	C	4.1.2.n	SP_1

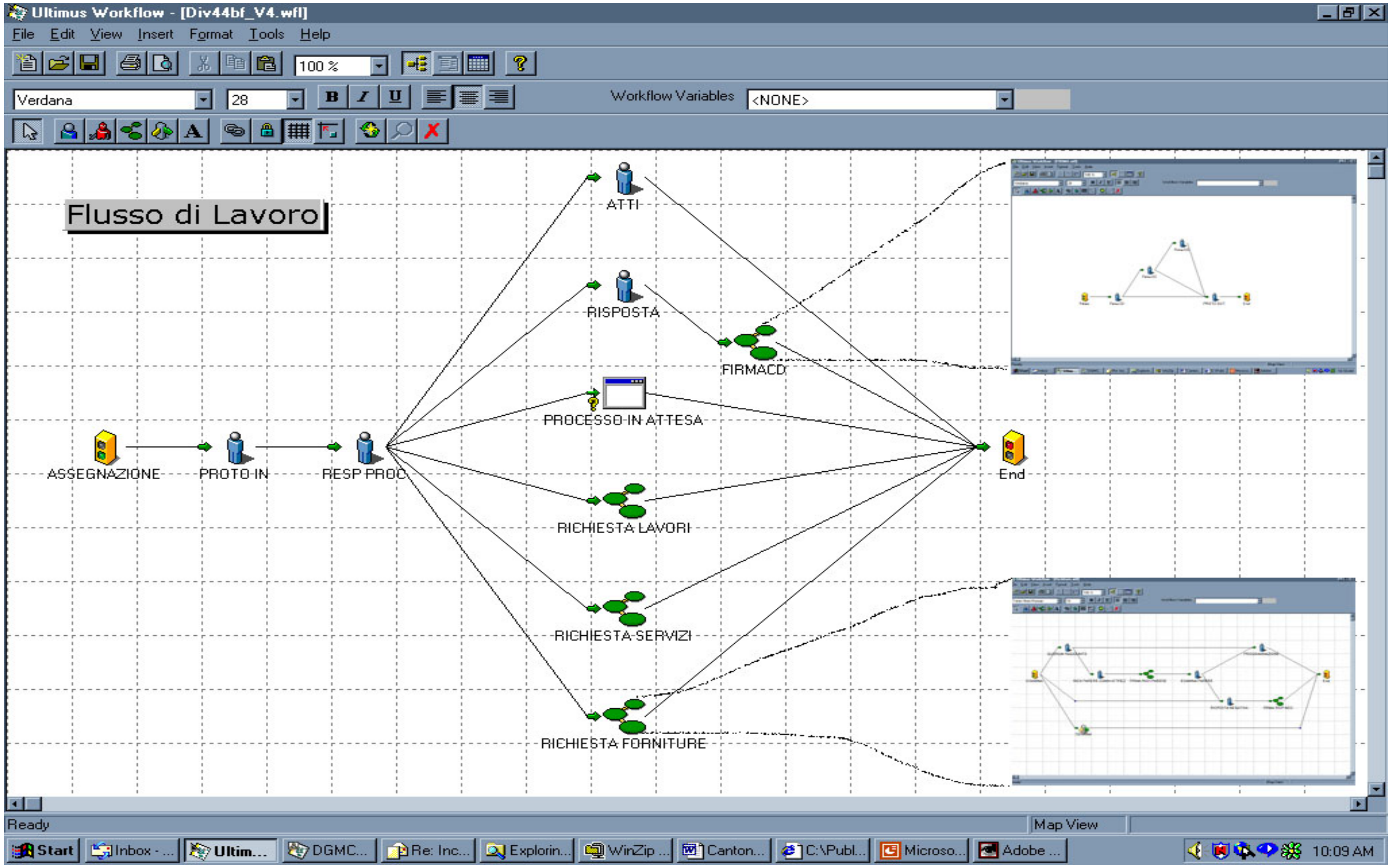
The Experiment Process: Characterization of the lab. projects, and the synthetic processes

Modeling Entity	No. before WA	No. after WA
SYSTEMS	**1**	**1**
SUB-SYSTEMS	**10**	**10**
ACTORS	**25**	**27**
• People	21	22
• Primary	8	8
• Secondary	13	14
• Systems	4	5
CLASSES	**65**	**70**
ASSOCIATIONS	**92**	**100**
• Inheritance	9	10
• Others	83	90
USE–CASES (full courses)	**24**	**40**
• Pilot Project	24	40
• Synthetic Processes	5	5

The Experiment Process: Involved Roles & Effort

Role	Effort (Man–months)
Project Manager	0.8
Business Process Analyst	3.5
Customer organization	0.4
Reference organization	0.3
WA Developer and Measurer	5.6
WA Application Verifier	1.4
Inspector	0.9
Observer & Editor	1.4

The Experimental Process: a Synthetic Process



Conclusions

Case study and experiment processes may lead the experimenter to start nested experiment processes.

This occurred when we were involved with testing hypothesis that related to high impact SW technology, which was still broadly modeled.