

# ***Metrics or Measures ? Which one would you better trust?***

**Alain Abran,**

***École de technologie supérieure,  
Université du Québec, Montreal, Canada***

**Universidad del Pais Vasco, San Sebastian,  
June 3, 2005**

# ***Objectives of the Presentation***

---

- ⊙ What do we know about measures?
- ⊙ What do we know about Software Metrics?
- ⊙ Software engineering and Measurements ?
- ⊙ Identification of the gaps where further research on software measurement is required

# *List of topics*

---

- ⊙ **Motivations and Objectives**
- ⊙ **Software Metrics**
- ⊙ **Models for Designing Measures**
- ⊙ **What is Generally Accepted?**
- ⊙ **International Standards**
- ⊙ **Conclusions and Future Work**

# Motivations

---

- ⊙ Measurement is fundamental in:
  - ❖ In day to day life
  - ❖ In business
  - ❖ In sciences and engineering
  
- ⊙ Measurement **instruments** are key to all
  
- ⊙ What is the status (*maturity*) of measurement in software?

# *Why do you measure?*

- ⊙ To understand:
  - ❖ To know – to learn
  
- ⊙ To plan:
  - ❖ To set targets
  
- ⊙ To control:
  - ❖ To monitor - To compare
  - ❖ To make adjustments

# ***Measurements***

Everywhere in sciences and engineering

- ❖ Where do they come from?
- ❖ How do you know that our measures are OK?

# *List of topics*

---

- ⊙ Motivations and Objectives
- ⊙ **Software Metrics**
- ⊙ Models for Designing Measures
- ⊙ What is Generally Accepted?
- ⊙ International Standards
- ⊙ Conclusions and Future Work

# ***Software Metrics***

The dominant view in software measurement =

**« Software Metrics »**



# ***Software Metrics***

- ⊙ Lines of code
- ⊙ McCabe
- ⊙ Function Points
- ⊙ Halstead
- ⊙ COCOMO
- ⊙ Software quality metrics: Hundreds +
- ⊙ Software complexity metrics: Hundreds +
- ⊙ OO metrics: Hundred ++
- ⊙ Estimates and Estimations models....

# *Quality of 'metrics'*

- ⊙ Do you get:
  - ❖ Reproducible results?
  - ❖ Repeatable results?
  - ❖ Accurate results?
  - ❖ Results that you can trust?
  
- ⊙ Who design these metrics?
  - ❖ How qualified are they?
  - ❖ Who verifies their metrics proposals?

# *Software Metrics*

How are they designed?

- ❖ Anything that can be 'counted'

⊙ How are they defined:

- ❖ Often labels
- ❖ An algorithm

⊙ How do we know if they are valid?

- ❖ Sometimes validation based on measurement theory
- ❖ Sometimes on claims of 'usefulness'

# *List of topics*

---

⊙ Motivations and Objectives

⊙ Software Metrics

⊙ **Models for Designing Measures**

⊙ What is Generally Accepted?

⊙ International Standards

⊙ Conclusions and Future Work

# ***Analytical Tools of Measurement***

---

- ◎ **Measurement Process model**

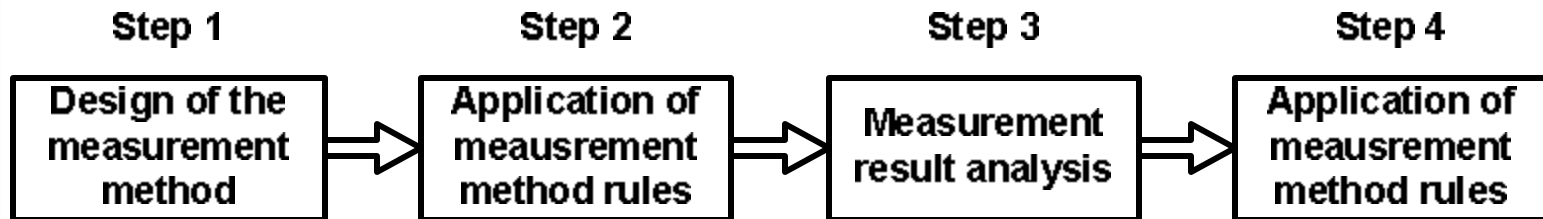
  - ↳ Abran & Jacquet

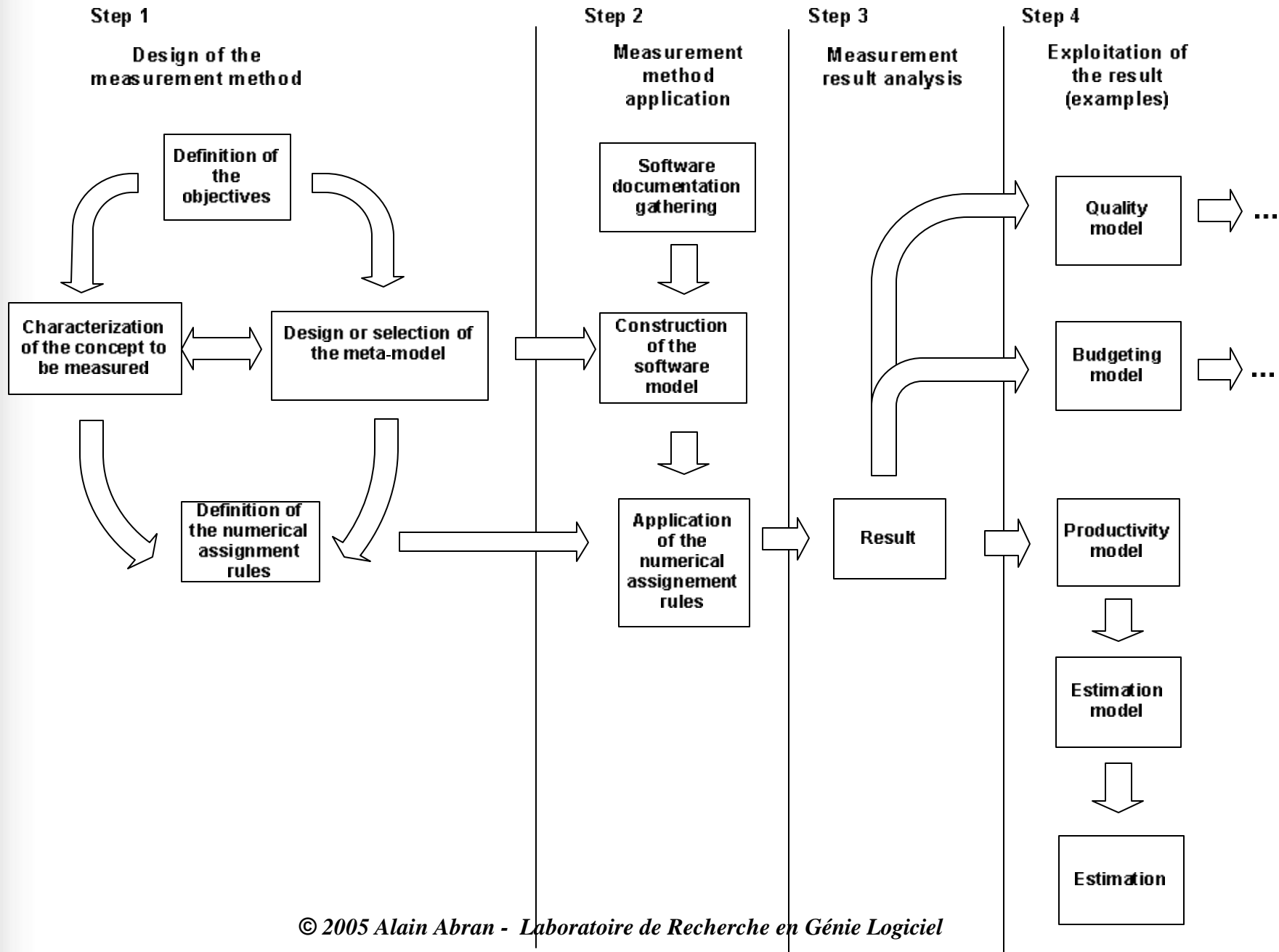
- ◎ **Metrology:**

  - ❖ **ISO International Vocabulary in Metrology**

# Measurement Process Model

## High-level measurement process model





# ***Metrology***

- ⊙ The long-standing international consensus on measurement terminology
  
- ⊙ The basis of the International System (IS) of measurements
  
- ⊙ National Metrology Agencies
  - ❖ The legal framework for weights and measures in industrialized countries



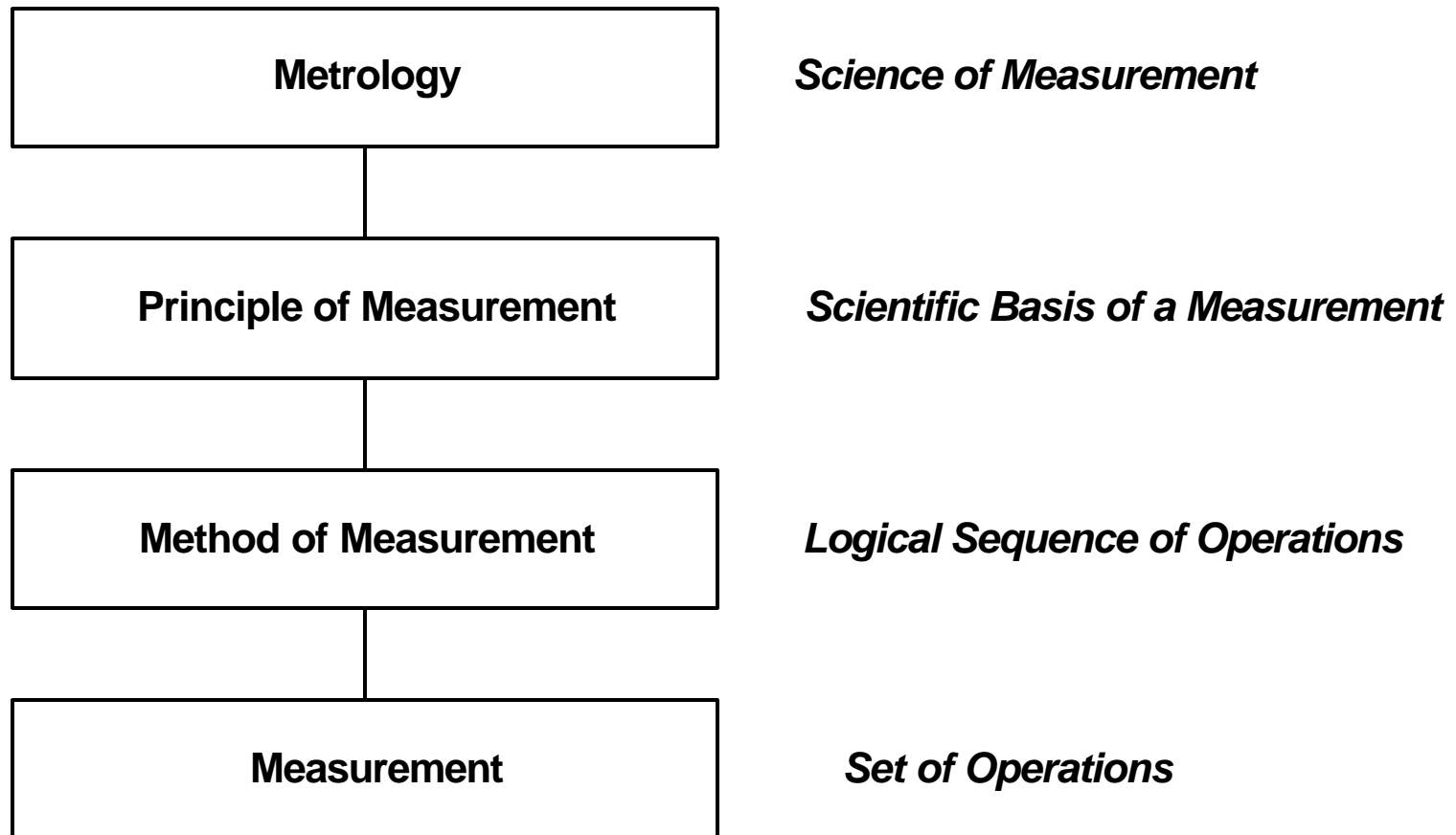
# ***Metrology Vocabulary***

- ⦿ Six categories of terms
  - ❖ + 120 terms
    - ↳ In increasing order of complexity!
  
- ⦿ Most challenging to grasp relationships across terms, understand, and remember!

# ***Metrology Model***

---

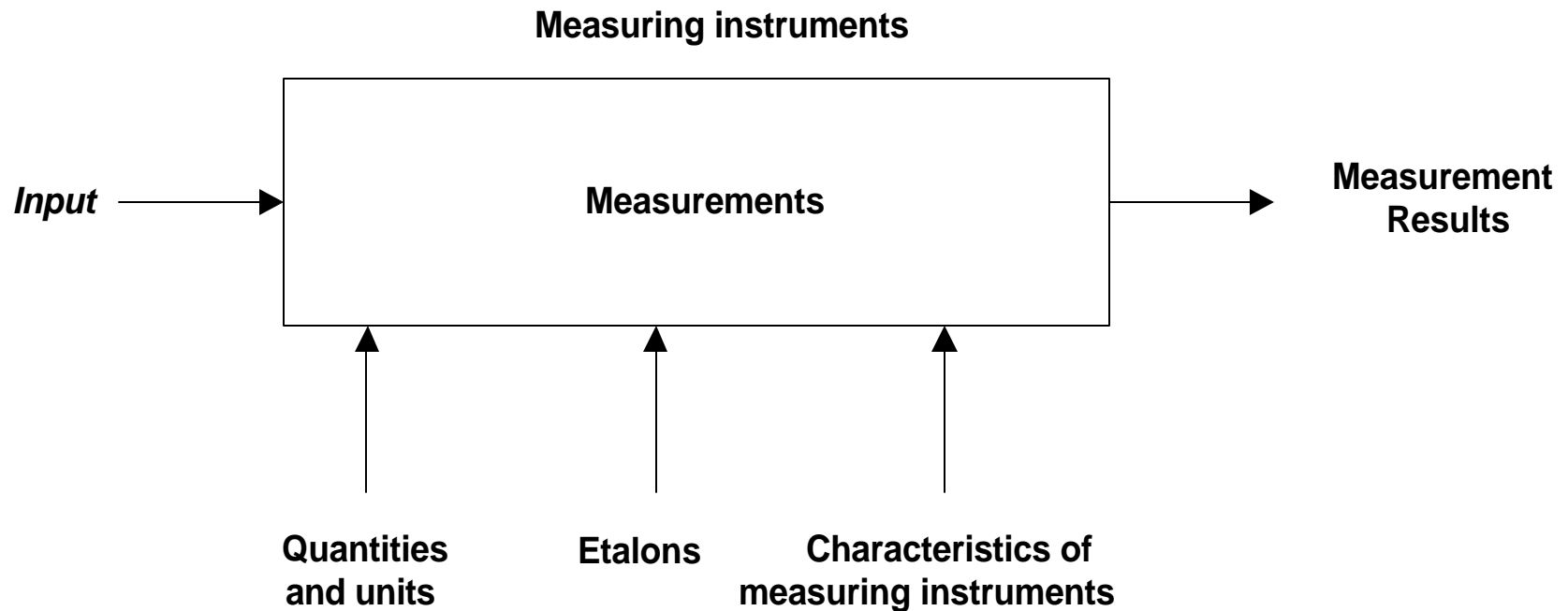
## Measurement foundations



# ***Metrology Model***

---

## High-level model of the ISO Vocabulary

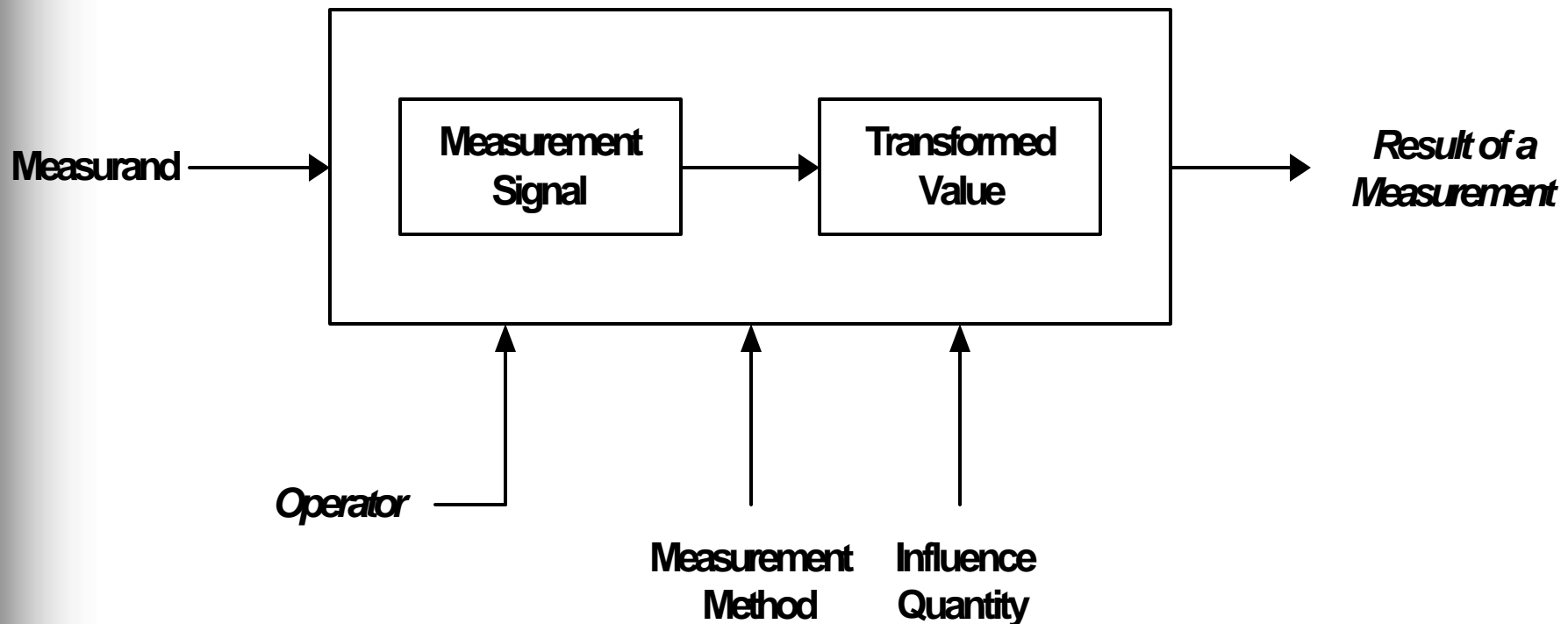


# ***Metrology Model***

---

## **Measurement Procedure**

### **Measurement Procedure**



# Metrology Model

## Measurement results

<i>Types of measurement results</i>	<i>Modes of verification of measurement results</i>	<b>Uncertainty of measurement</b>
<b>Indication (of a measuring instrument)</b>  <b>Uncorrected result</b>  <b>Corrected result</b>	<b>Accuracy of measurement</b>  <b>Repeatability (of results of measurements)</b>  <b>Reproducibility (of results of measurements)</b>	<b>Experimental standard deviation</b>  <b>Error (of measurement)</b>  <b>Deviation</b>  <b>Relative error</b> <b>Random error</b> <b>Systematic error</b>  <b>Correction</b> <b>Correction factor</b>

# *Mapping between models*

---

## Alignment of metrology concepts with the measurement process model

Measurement process model	Design of measurement methods	Application of measurement method rules	Measurement results analysis	Exploitation of measurement results
ISO metrology model	Quantities and units	Measuring instruments  Characteristics of measuring instruments	Measurement results	

# *List of topics*

---

- ⊙ Motivations and Objectives
- ⊙ Software Metrics
- ⊙ Models for Designing Measures
- ⊙ **What is Generally Accepted?**
- ⊙ International Standards
- ⊙ Conclusions and Future Work

# ***SWEBOK***

- ⦿ Generally Accepted:
  - ❖ What applies most of the time, to most projects, and which value has been validated by the community of peers
    - ↳ Project Management Institute
  
- ⦿ Software Engineering Body of Knowledge - SWEBOK



# Measurement within SWEBOK

SWEBOK KA	Topics	Step 1 Design	Step 2 Measuring	Step 3 Results	Step 4 Uses
<b>Software engineering requirements</b>	Process support and management				×
	Requirements negotiation				×
	Document quality				×
	Acceptance tests				×
	Requirements tracing				×
<b>Software engineering design</b>	Measures			×	
<b>Software engineering testing</b>	Evaluation of the program under test				×
	Evaluation of the tests performed				×
<b>Software engineering maintenance</b>	Software Maintenance Measurement				×

# Measurement within SWEBOK

SWEBOK KA	Topics	Step 1	Step 2	Step 3	Step 4
<b>Software configuration management</b>	Surveillance of software configuration management				×
<b>Software engineering management</b>	Goals				×
	Measurement selection				×
	Measuring software and its development				×
	Collection of data		×		
	Software Measurement Models			×	
<b>Software engineering process</b>	Methodology in process measurement		×		
	Process Measurement Paradigms				×

# Measurement within SWEBOK

---

SWEBOK KA	Topics	Step 1	Step 2	Step 3	Step 4
<b>Software engineering quality</b>	Measuring the value of quality				×
	Fundamentals of Measurement	×			
	Measures			×	
	Measurement analysis techniques				×
	Defect characterization				×
	Additional Uses of SQA and V&V data				×

# ***Generally accepted knowledge about software measurement?***

Strong recognition of benefits:

- ⊙ to understand, plan, monitor and control
- ⊙ Foundations = ??
- ⊙ And little metrology strengths

# *List of topics*

---

- ⊙ Motivations and Objectives
- ⊙ Software Metrics
- ⊙ Models for Designing Measures
- ⊙ What is Generally Accepted?
- ⊙ **International Standards in Software Measurement**
- ⊙ Conclusions and Future Work

# ***International Standards***

Currently, only for:

- ⦿ Software Products Quality
- ⦿ Software Functional Size

# *How to Measure Software Quality?*

ISO 9126 on Software **Products** Quality

- ⊙ Part 1: Quality Models and Definitions
  
- ⊙ Parts 2 to 4: **Technical Reports**
  
- ⊙ Over + 120 Metrics !
- ⊙ with little about:
  - ❖ measurement method for each (labels & algorithms)
  - ❖ Validity & Quality of these 'metrics' ??
  
- ❖ Then (if used in a non consistent manner), how do figure out how measurement results compare across contexts, across time, and across measurers?

# ***Software Functional Size***

How do you measure software size?

- ⊙ The technical size = ?
- ⊙ The functional size = ?



# *Functional Size*

A unique set of measures in software engineering:

- ⊙ Designed in the late 1970's:
  - ❖ By Albrecht, from IBM, using 24 MIS projects
  
- ⊙ Published in the early 1980's
- ⊙ User group in the mid 1980's
  - ❖ Measurement Manual
  - ❖ Training & Certification

# *Functional Size*

Does it withstand the test of time?

- ⊙ Still in use and referenced

But

- ⊙ The basic method has not evolved significantly since the early 90's
- ⊙ Software has changed considerably
- ⊙ Outside of MIS domain = ?
- ⊙ In the early 90's: + 30 variations to tackle weaknesses

# ***Functional Size***

Innovation in software measurement:  
Standardization through ISO

First a meta-standard to layout the ground rules  
about functional size measurement: ISO 14143

- ⊙ Part 1 = Definitions of Key Concepts
- ⊙ Part 2 = Conformity Assessment
- ⊙ Part 3 = Verification Guide
- ⊙ Part 4 = Set of References
- ⊙ Part 5 = Functional Domains

# ***Functional Size***

Four specific methods approved by ISO

- ⊙ ISO 19761: COSMIC-FFP
- ⊙ ISO 20926: IFPUG
- ⊙ ISO 20968: MKII
- ⊙ ISO 24570: NESMA

Will they withstand the test of time as measurement methods?

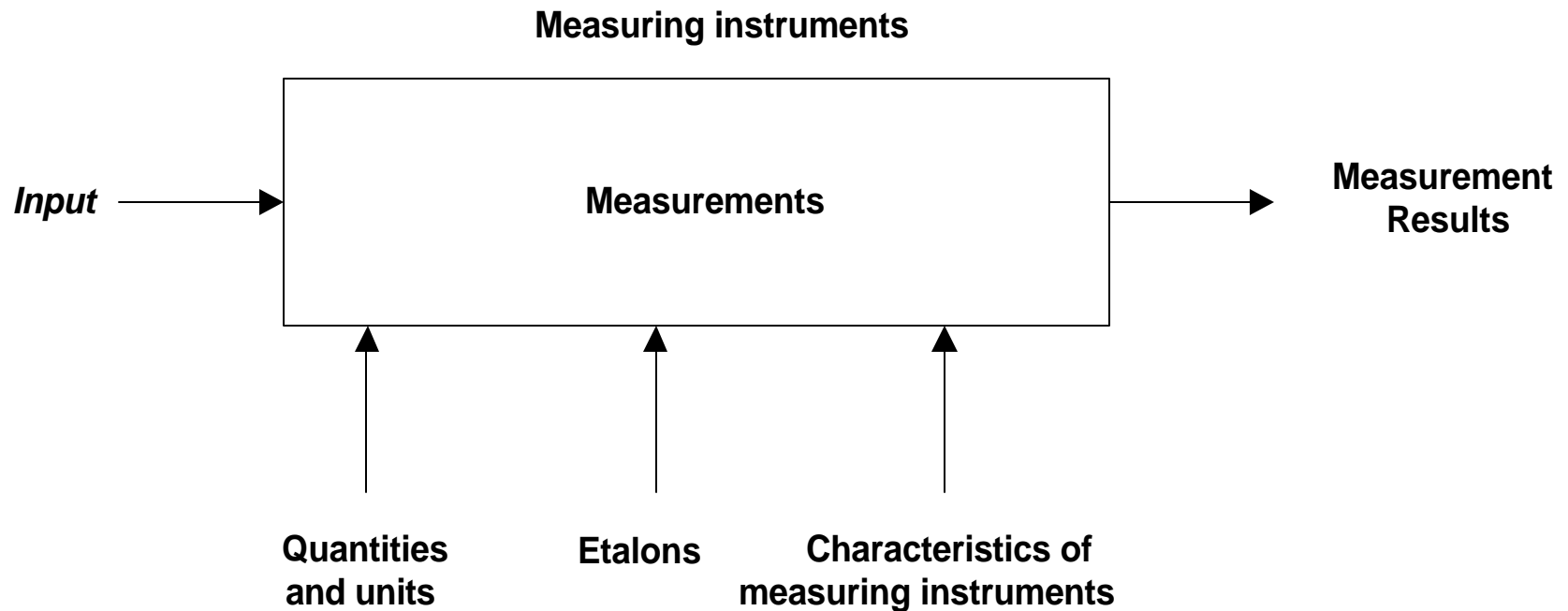
Will a consensus emerge?

Do they meet 'metrologoy' criteria?

# ***Metrology Model***

---

## **High-level model of the ISO Vocabulary**



# Metrology Model

## Measurement results

<i>Types of measurement results</i>	<i>Modes of verification of measurement results</i>	<b>Uncertainty of measurement</b>
Indication (of a measuring instrument)  Uncorrected result  Corrected result	Accuracy of measurement  Repeatability (of results of measurements)  Reproducibility (of results of measurements)	Experimental standard deviation  Error (of measurement)  Deviation  Relative error Random error Systematic error  Correction Correction factor

# *List of topics*

---

- ⊙ Motivations and Objectives
- ⊙ Software Metrics
- ⊙ Models for Analytical Tools of Software Measurement
- ⊙ What is Generally Accepted?
- ⊙ International Standards
  
- ⊙ **Conclusions and Future Work**

# ***Summary & Conclusion***

---

**Generally accepted knowledge about Measurement in software:**

- ⊙ **Extensive set of references on the use of measurement results in assessment and predictive models.**
  
- ⊙ **Little discussion on:**
  - ❖ **Quality of measurement results**
  - ❖ **Quality of measuring instruments**
  
- ⊙ **Limited knowledge on the design of measurement methods**



# ***Conclusions - Next***

- ⊙ The field of « software metrics » is going where after 30 years of research?
  - ❖ Why not learn from the masters in measurement?
- ⊙ Most majors R & D contributions still waiting for you!
- ⊙ Standardization is critical for measurement!

# *Questions*

?