第三届（2018）

中国软件成本度量大会

暨 IWSM-MENSURA INTERNATIONAL CONFERENCE

Estimation of Software Projects: Placebos or Engineering Practices?

演讲人：Alain Abran

Estimation of Software Projects: Placebos or Engineering Practices?

Alain Abran

SSM & IWSM⁄ MENSURA conferences
Beijing (China) September 19, 2018

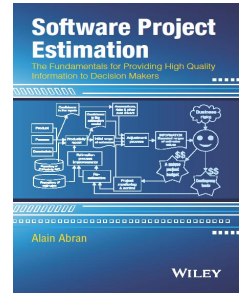
Presenter background - Alain Abran

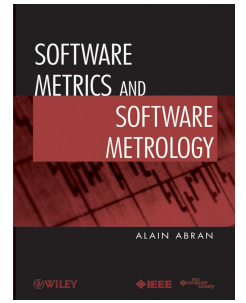
20 years

+ 20 years

45 PhD

ISO: 19761, 9126, 25000, 15939, 14143, 19759





- **Development**
- **Maintenance**
- **Process Improvement**

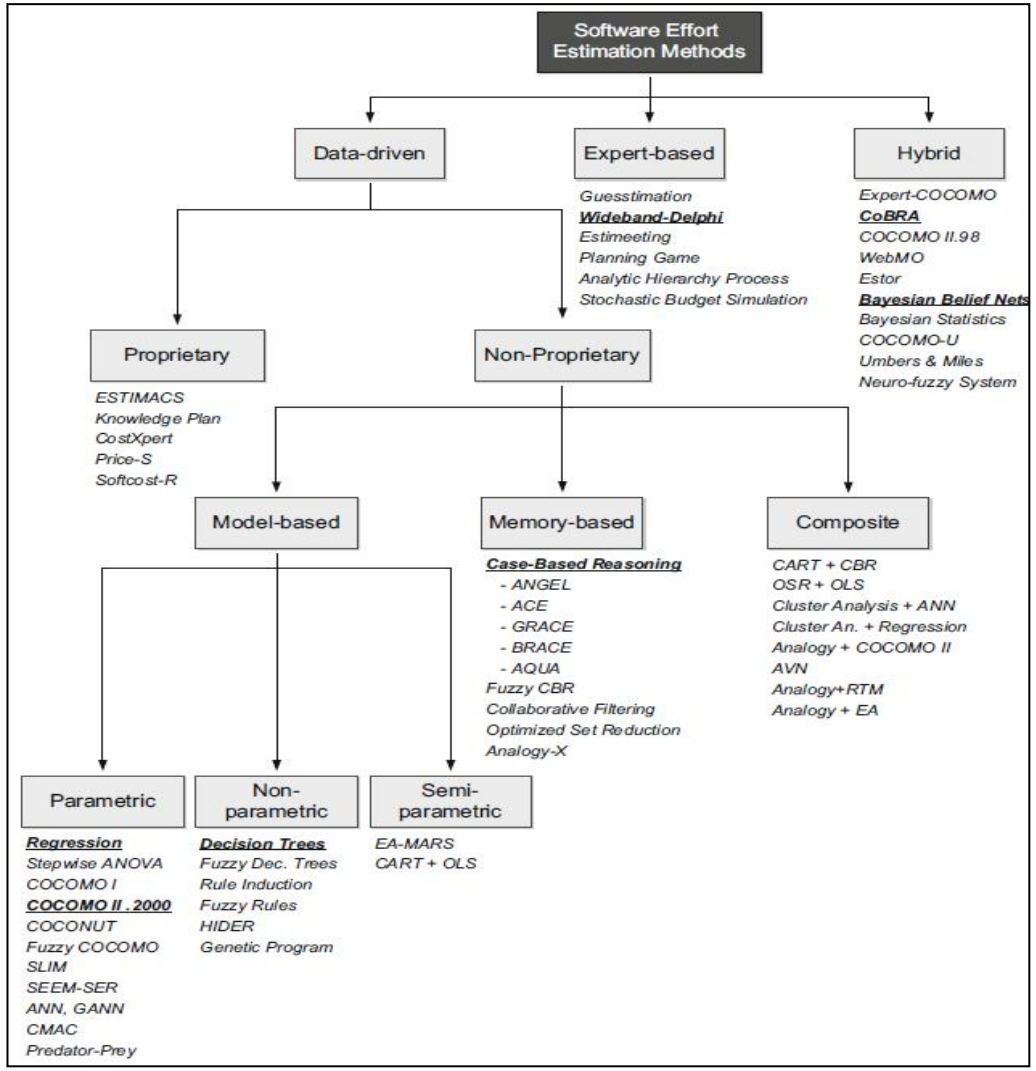
Agenda

1. Complex Effort Estimation Models: Placebos?
2. A key concept for estimation: simple productivity-based estimation models for monitoring & control
3. Size estimation: The foundation for Effort Estimation

Agenda

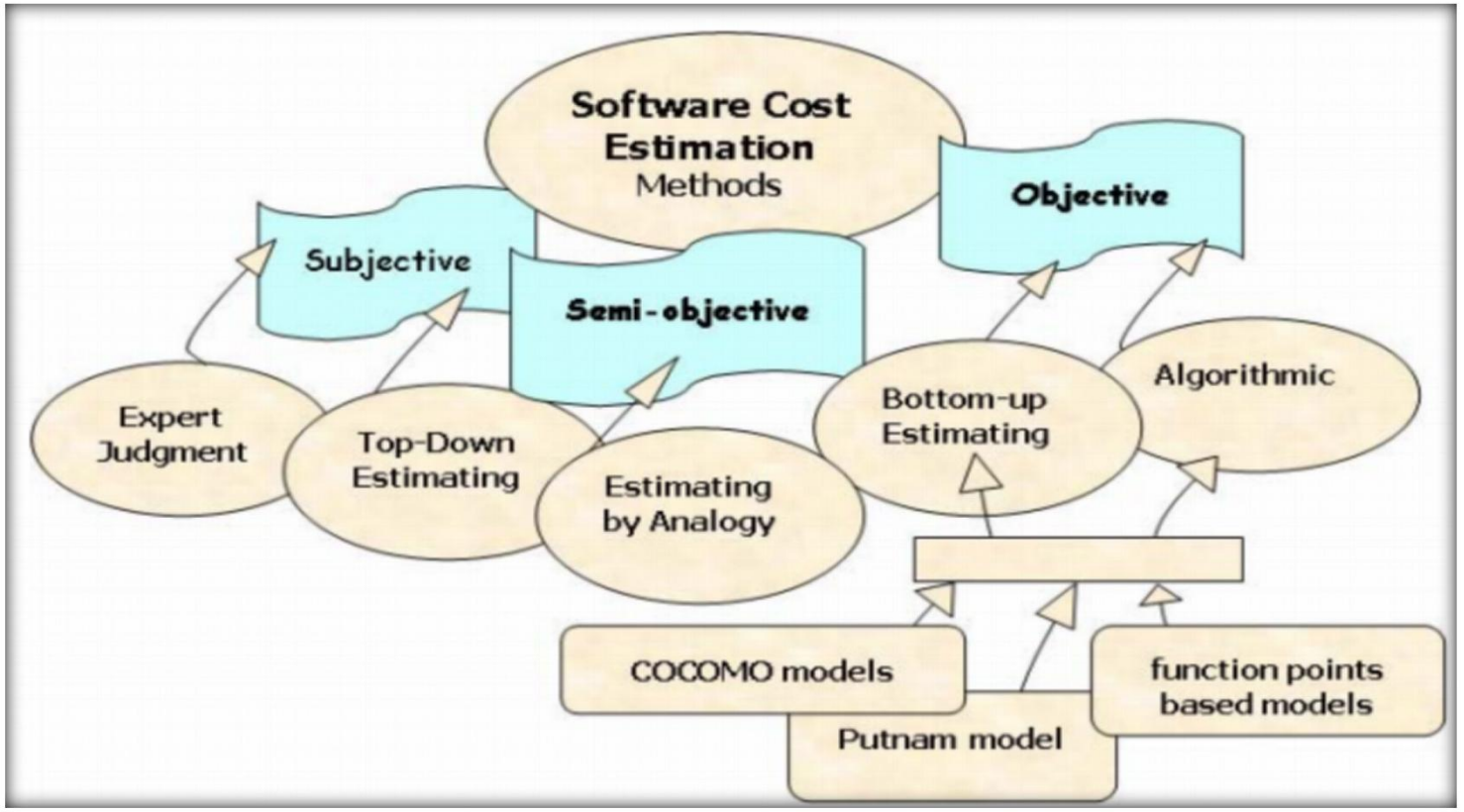
1. **Complex Effort Estimation Models: Placebos?**
2. A key concept for estimation: simple productivity-based estimation models for monitoring & control
3. Size estimation: The foundation for Effort Estimation

Complex White box & Black Box Effort Estimation Models



Source: Adam Trendowicz, Ross Jeffery. *Software Project Effort Estimation*. Springer, 2014.

Black Boxes & White Boxes Estimation Models



Source: Violeta Bozhikova, Mariana Stoeva. An Approach for Software Cost Estimation. *CompSysTech'10*, June 17–18, 2010, Sofia, Bulgaria, pages 119-124.

Examples of major weaknesses in Estimation Models

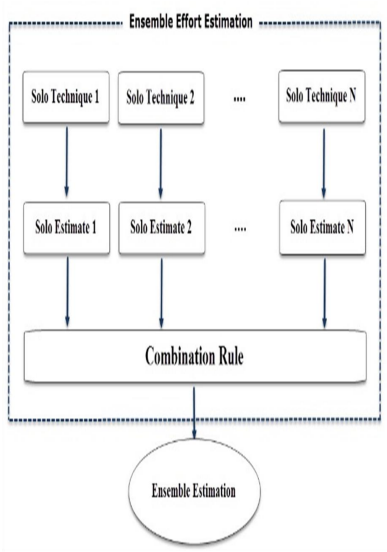
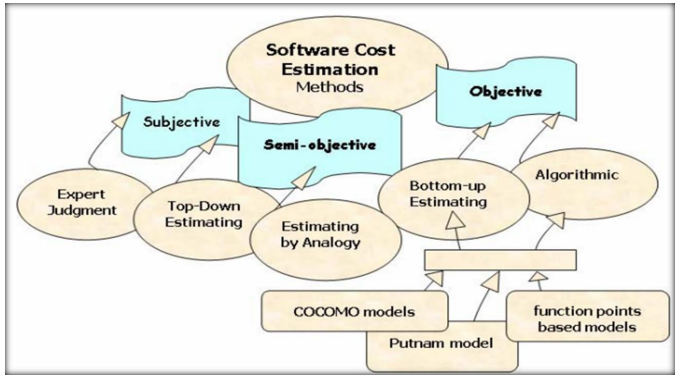


Fig. 1. Ensemble Effort Estimation (EEE) process.



$$\text{Impact of all cost drivers} = \sum_{n}^{i} PF_i$$

$$\text{Effort} = a \times \text{Size} \times \left(\sum_{n}^{i} PF_i\right) + b$$

Sources:

- 'Improved Estimation of Software Development Effort Using Classical and Fuzzy Analogy Ensembles', Idri, Hosni, Abran, Applied Soft Computing, Elsevier, vol. 49, 2016.
- 'On the value of parameter tuning in heterogeneous ensembles effort estimation', Hosni, Idri, Abran, Bou Nassif. In Soft Computing, Springer, 30 November 2017, pp. 1-34

Key Weaknesses in COCOMO-like Estimation Models

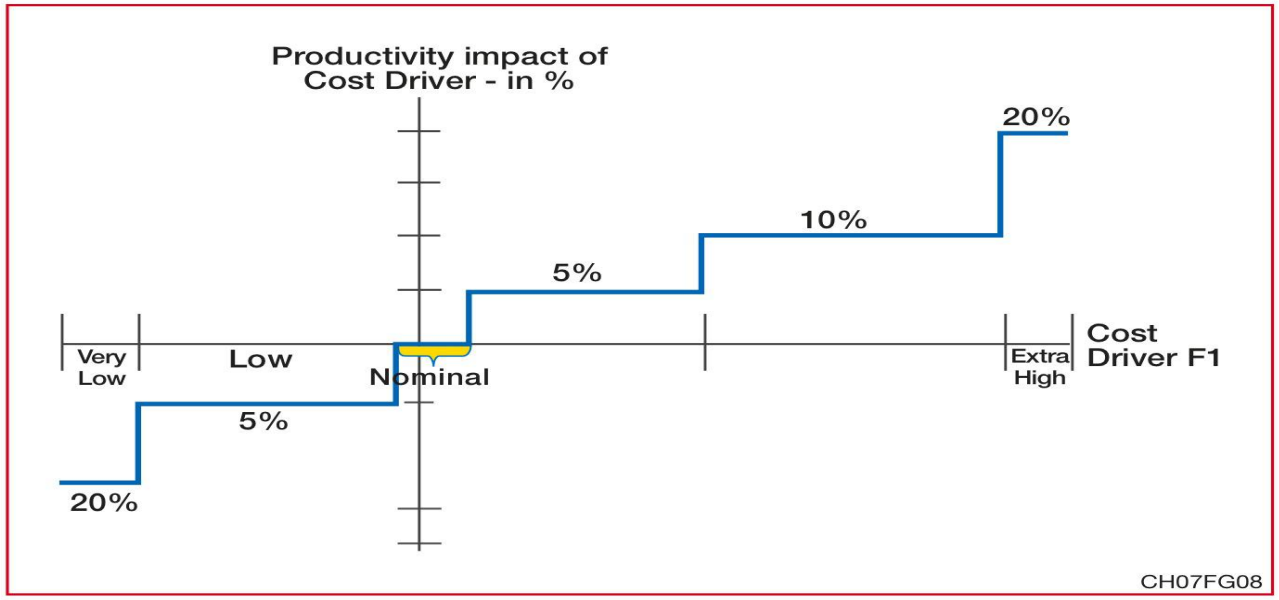


Figure 7.8 A step-function estimation model-with irregular intervals.

Impact guessed by 'experts'

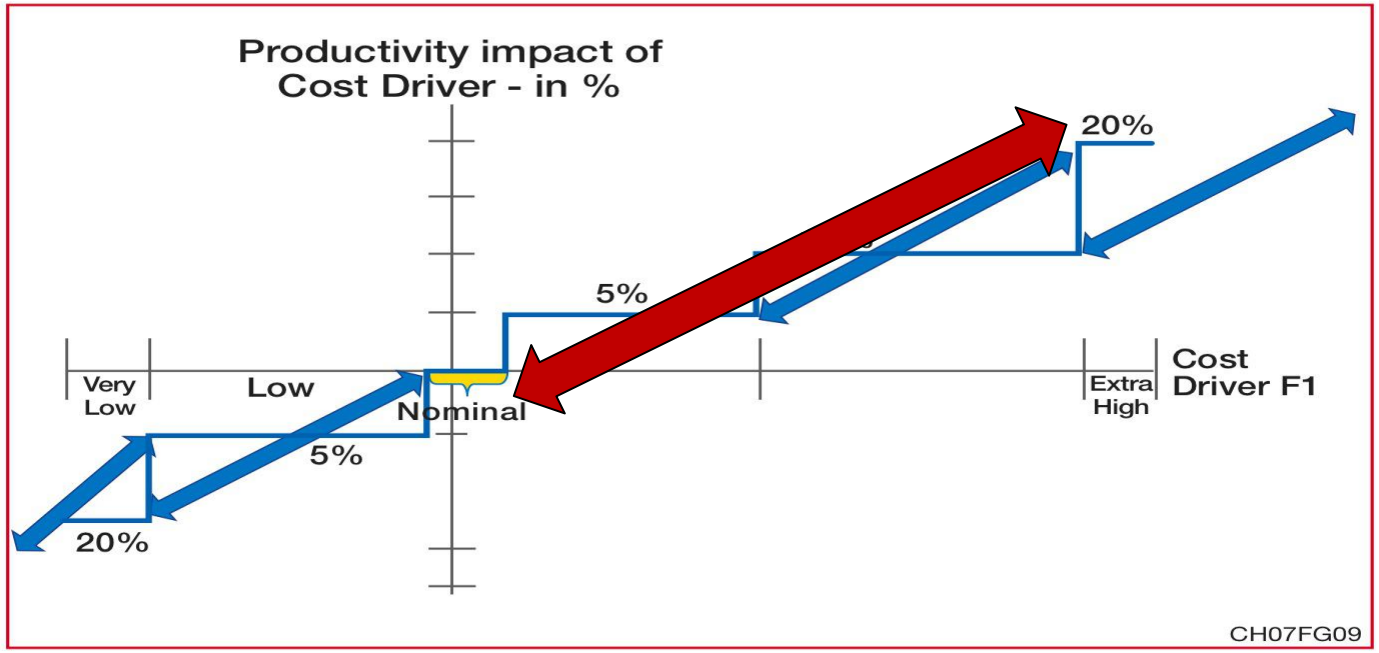
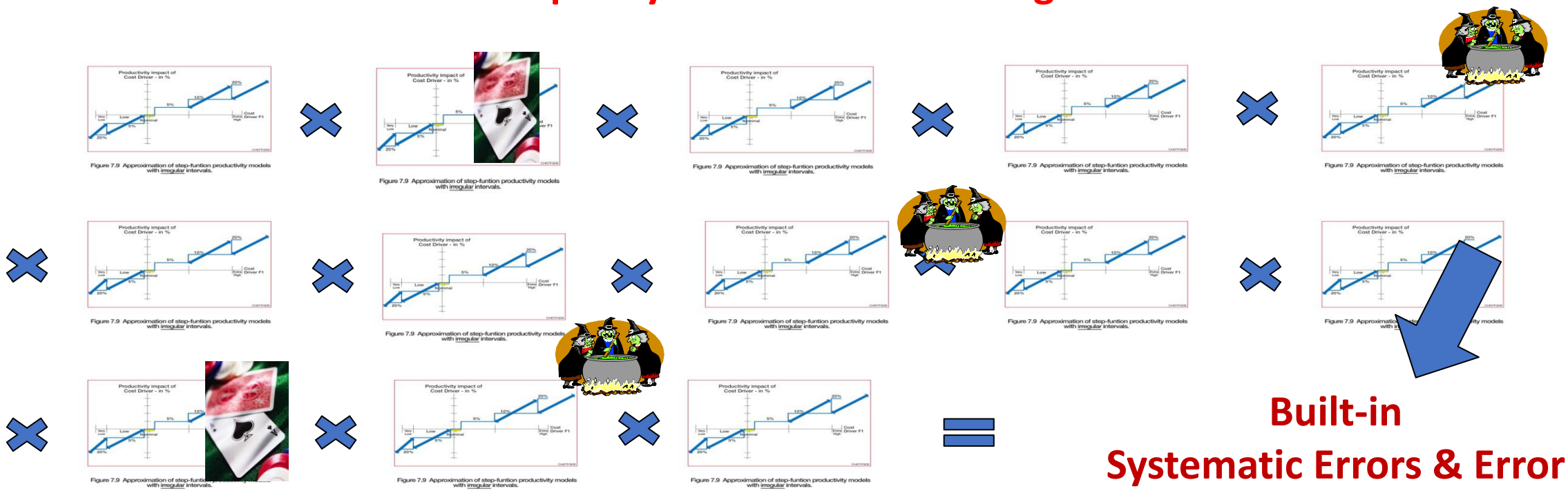


Figure 7.9 Approximation of step-funtion productivity models with irregular intervals.

Each COCOMO cost driver
=

an estimation
sub-model
with unkown quality &
large errors

COCOMO-like estimation models:
Effort is a function of (Size & +17 step-functions)
of unknown quality combined into a single number!



Built-in Systematic Errors & Error Propagation

M.I.T. study on COCOMO81 (Kemerer, 1987)

Small scale replication study - 17 projects

	Basic Exponential on Size	Intermediate & 15 cost drivers	Detailed & 4 project phases
R^2 (max=1.0)	**0.68**	**0.60**	**0.52**
Model Errors **(Mean magnitude of relative errors - MMRE)**	**610%**	**583%**	**607%**

Estimation Outcomes!

The ***'feel-good'***,

Quick & Easy…

Examples of similar ‘feel-good’ inputs to Estimation Models

- Story Points

- IFPUG SNAP points (for Non Functional Requirements):
  - Factors improperly bundled into a single number as adjustments to the Size variable:

Planning Poker & Story Points in Agile: Do they meet measurement criteria?

- Repeatability:
  - different individuals, in different contexts, at different times, & following the same measurement procedures will **NOT** obtain the same measurement results.
- Measurement results:
  - obtained with minimal judgment.
  - results auditable.

FAIL

FAIL

FAIL

Story Points

- **Unaccountability…..!**

Non-Functional Requirements: IFUG **SNAP Points**

Category	Sub-concepts for the classification	SNAP weights basis
Data Entry Validation	Nesting level complexity	2,3,4 * number of DETs
Logical operations	Control flow complexity	4,6,10 * number of DETs
Mathematical operations	Control flow complexity	4,6,10 * number of DETs
Data formatting	Transformation complexity	2,3,5 * DETs
Internal data movements	Internal boundaries crossed DET transferred	5* (# of internal boundaries crossed)+2*(#DETs)
Functionality by data config.	Complexity	3,4,6 * Number of records
UI Changes	UI types complexity	2,-,4 * DETs
Help methods	Help types	1,2,3 * number of Help items
Multiple input methods	Media types	3,4,6 * number of controls
Multiple output methods	Media types	3,4,6 * number of controls
Multiple platforms	No. of platforms to operate	8 * Number of platforms
Database technologies	Level & type of normalization of the physical schema	1,3,4,5,7 * levels of normalization
System configuration		SP=(middleware config.)+2*(# backend config.) +3*(# interface config.)
Batch processing	Number of batches or transactions	2*(number of batches or transactions)
System critical (real-time)	- Type of transactions - No. critical trans.	5,10,15 * number of critical transactions
Component based software	Type of components (In-house reuse or 3rd party component	4,8 * number of unique components
Design complexity	Interface complexity	8,16,24 *# of COTS applications + 12,24,36 *#nonCOTS applications

Scale type		Admissible Transformation	Operations	Examples
Nominal	(R,=)	f unique	Name, distinguish	Colors, shapes
Ordinal	(R,>=)	f strictly increasing monotonic function	Rank, Order	Preference, hardness
Interval	(R,>=,+)	f(x)= ax + b, a>0	Add	Calendar time, temperature (degrees Celsius)
Ratio	(R,>=,+)	f(x)= ax, a>0	Add, multiply, divide	Mass, distance, absolute temperature (degrees Kelvin)
Absolute	(R,>=,+)	f(x)= x	Add, multiply, divide	Entity count

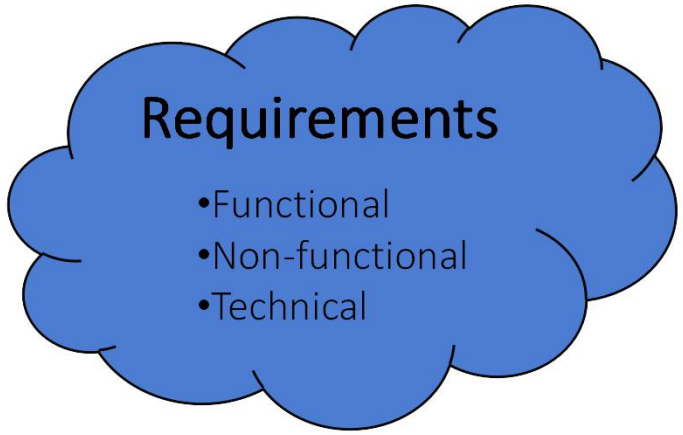
It fails primary school maths!

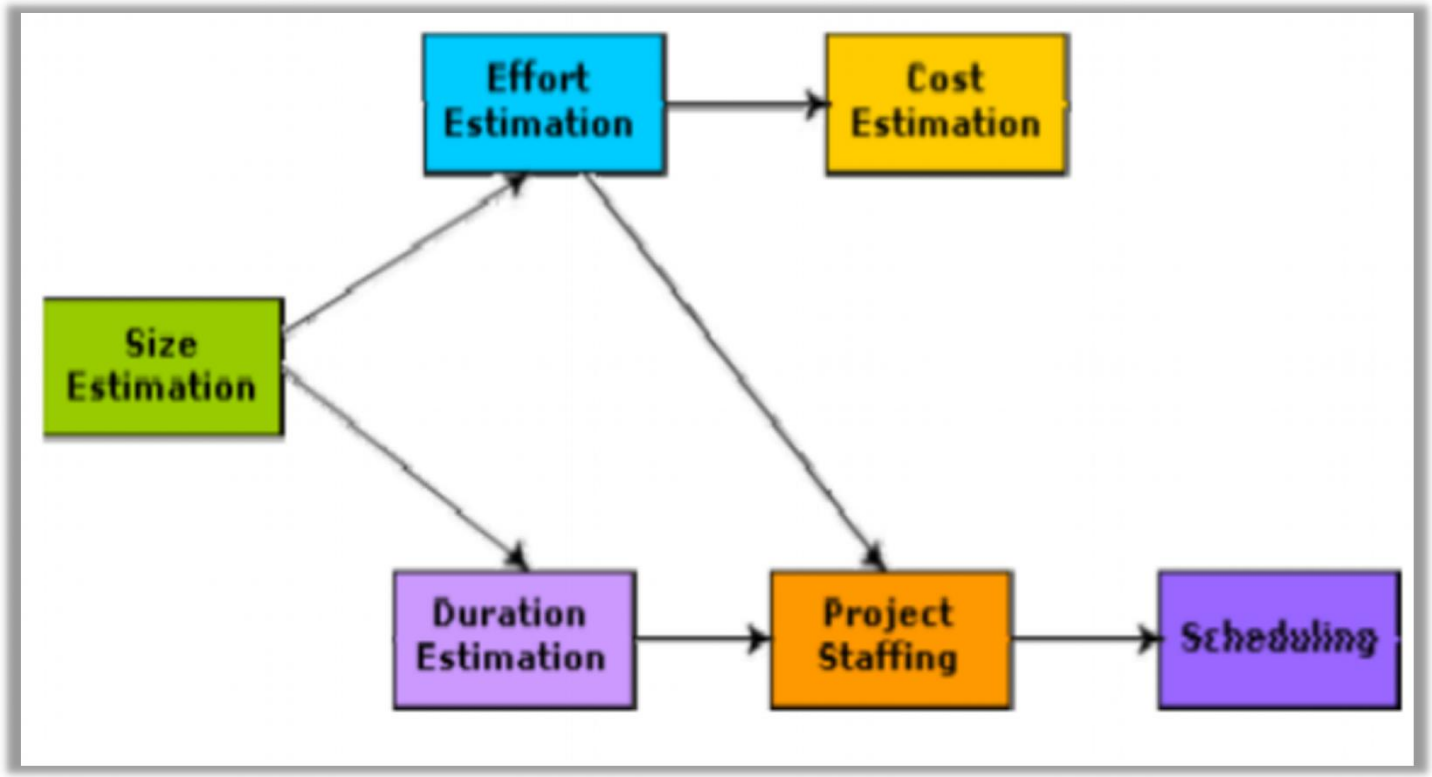
Feel-Good Placebos

Agenda

1. Complex Effort Estimation Models: Placebos?
2. **A key concept for estimation: simple productivity-based estimation models for monitoring & control**
3. Size estimation: The foundation for Effort Estimation

The general estimation process

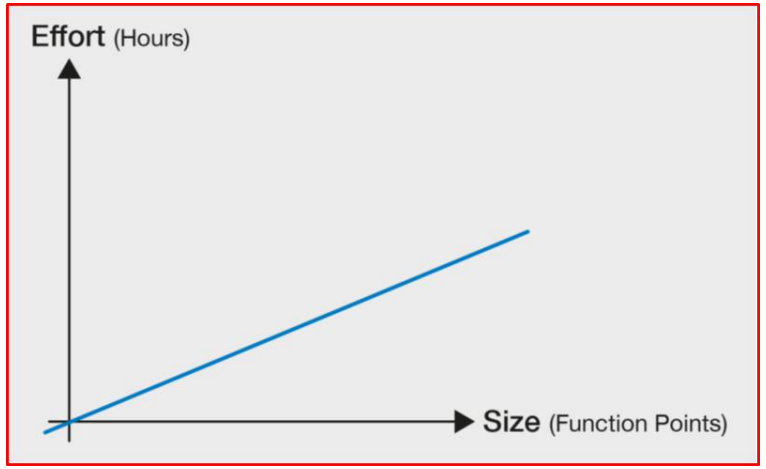




Source: Adapted from [Kharagpur 2010]

Productivity

Productivity = Outputs / Inputs

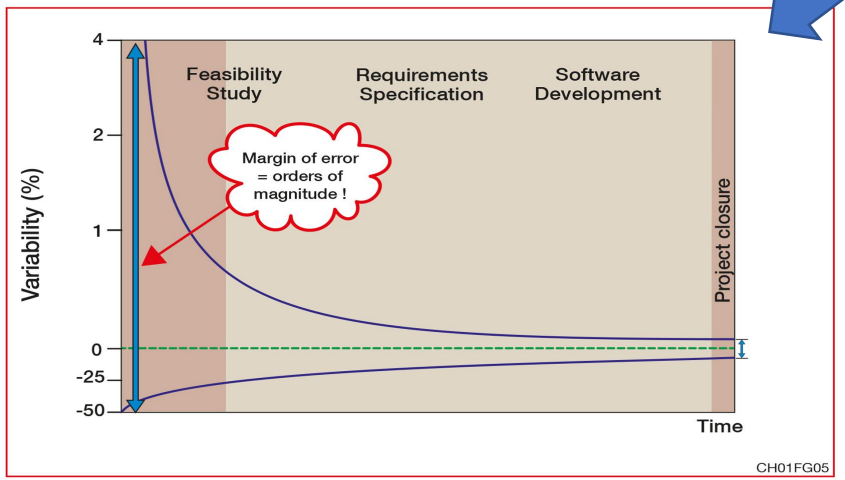


Number of functions delivered
Size in CFP, FP, LOC

Number of worked hours
(Effort in PM or PersonHour)

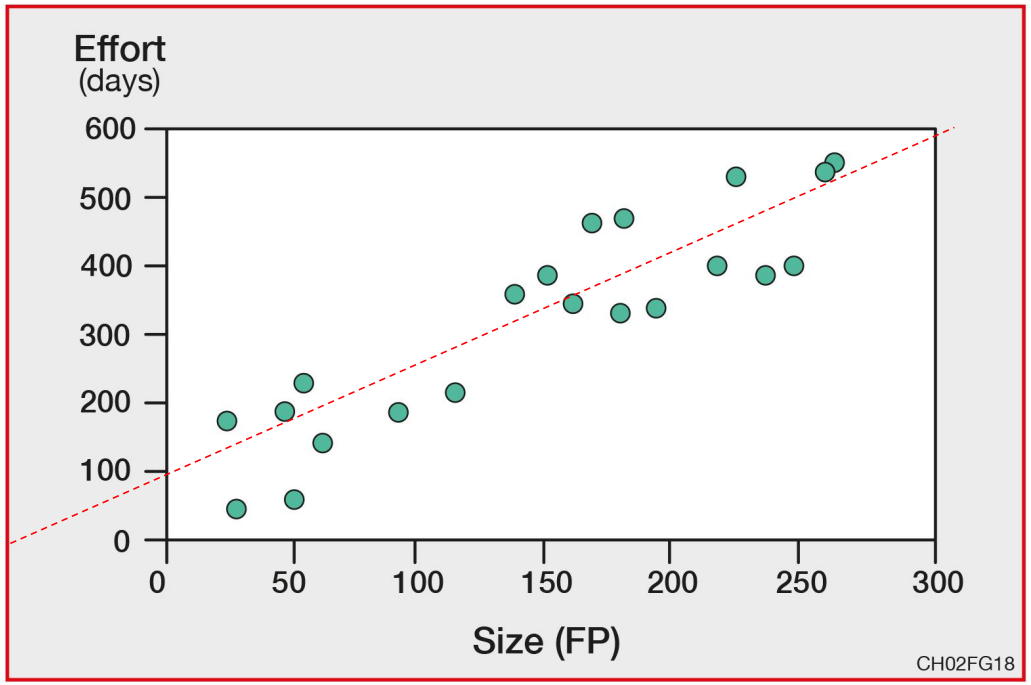
Example: A team developed a software with a size of 100 COSMIC Function Points (CFP) in 5 person month

Productivity = 100 CFP / 5 PersonMonth = 20 CFP/PersonMonth



Organization Data Repository

Software Size as the dominant factor well-managed organizations

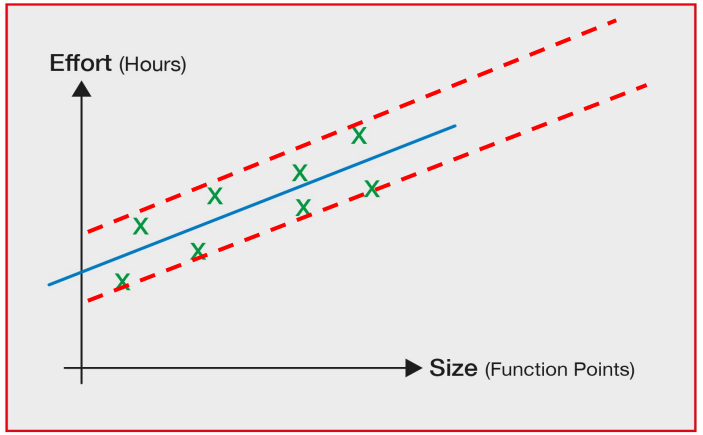


Homogeneous dataset of 21 projects (Abran 1994)

Software Size versus all Other Factors

In well-managed organizations:

- ✓ The increases in functional size explain 80% to 90% of the increases in effort.
  - ➢ All other factors together explain only 10% to 20% of the variations.

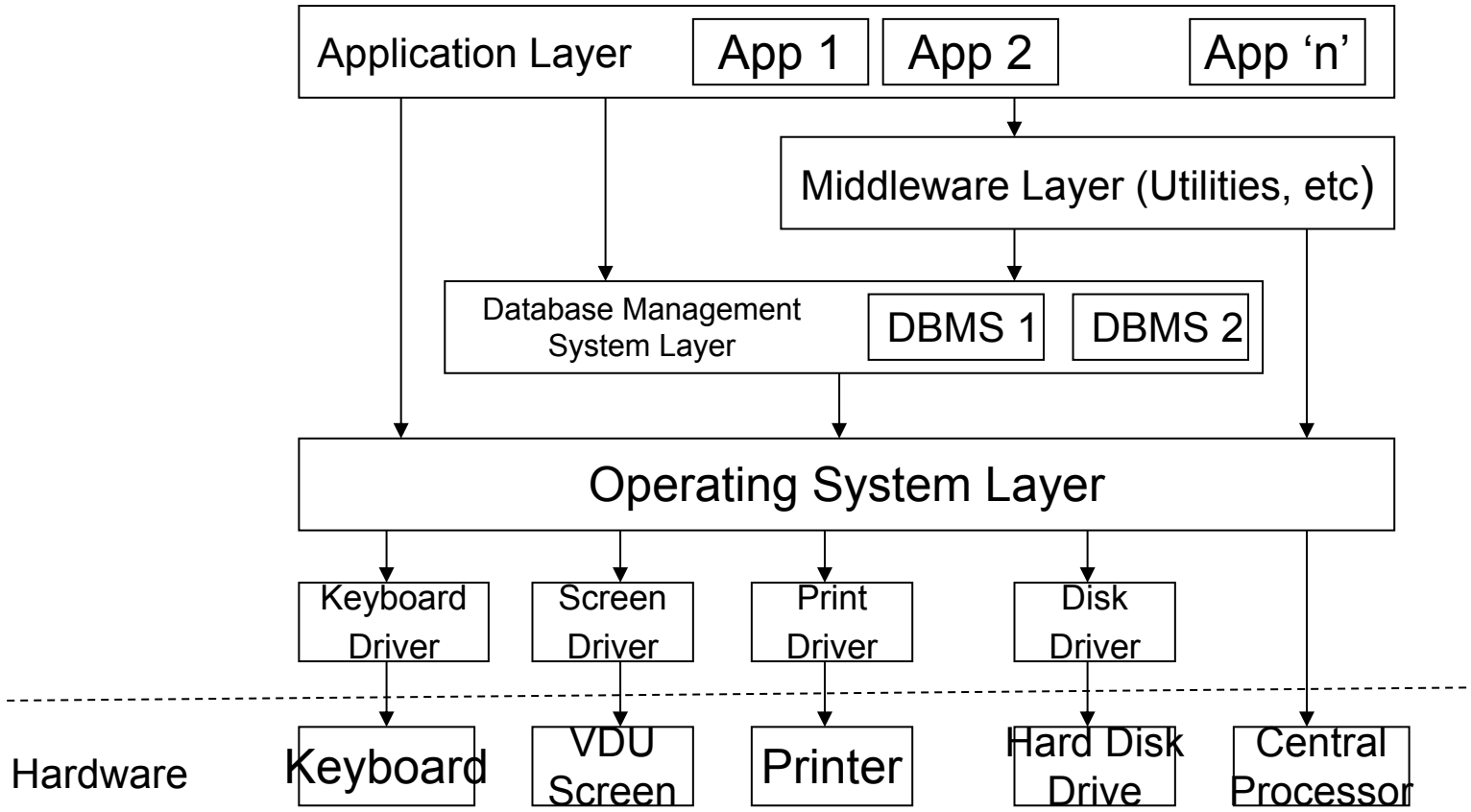


Examples of other factors:

External Interfaces

Non-Functional & Quality

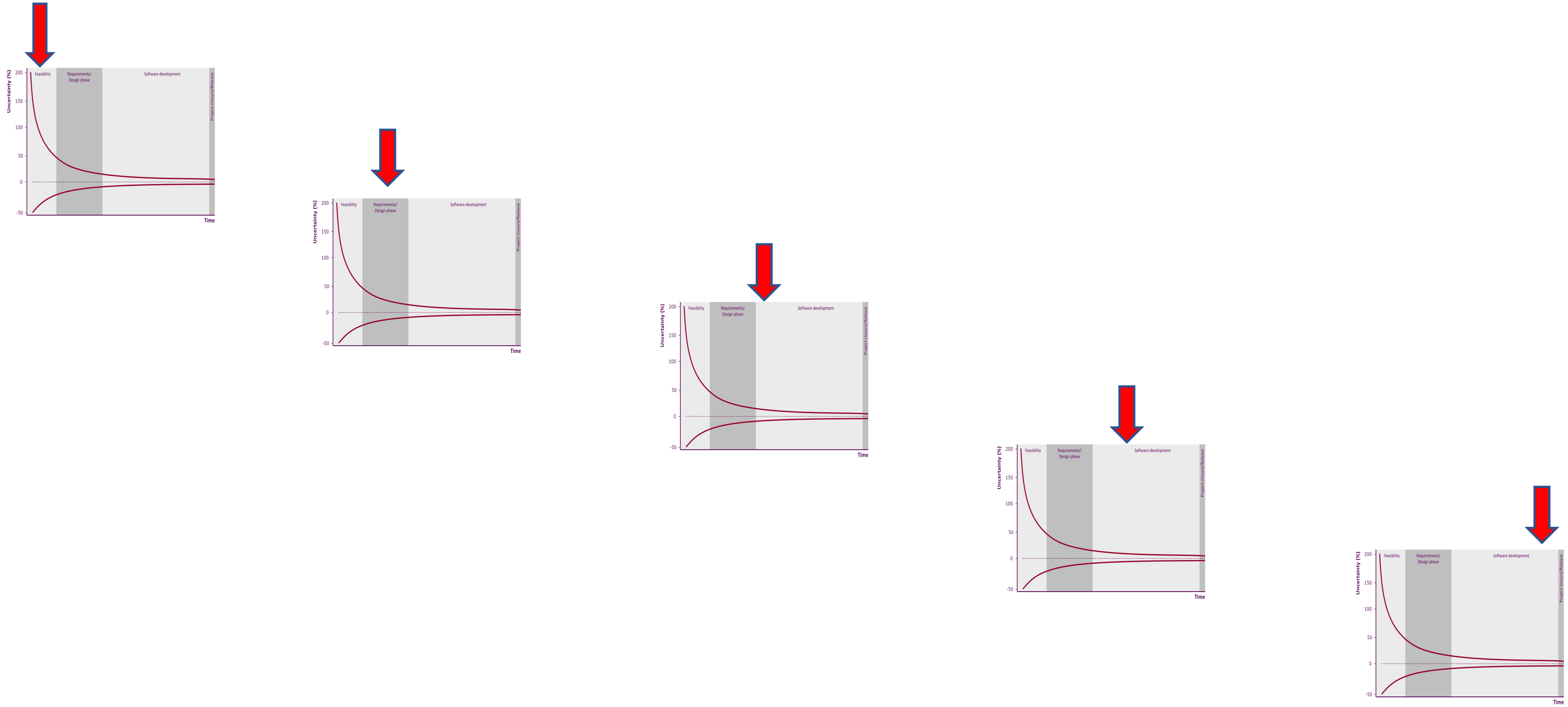
1. Usability requirements
2. Performance requirements
3. Logical database requirements
4. Design constraints
5. Standards compliance
6. Software system attributes



Agenda

1. Complex Effort Estimation Models: Placebos?
2. A key concept for estimation: simple productivity-based estimation models for monitoring & control
3. **Size estimation: The foundation for Effort Estimation**

Size & Estimation throughout the lifecycle



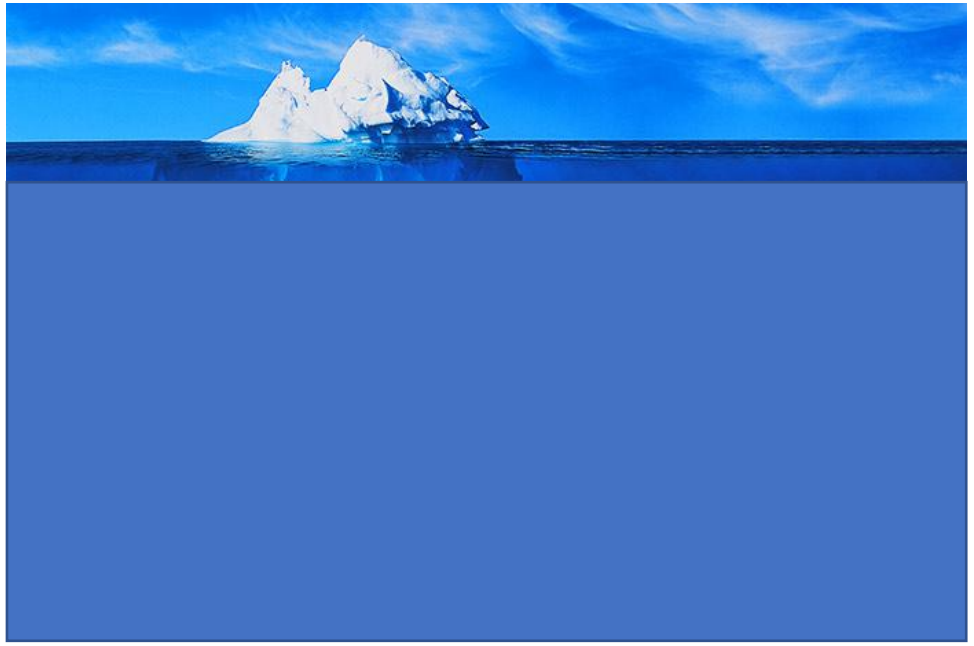
What Size is VISIBLE at Feasibility?

INTERNATIONAL STANDARD **ISO/IEC/ IEEE 29148**
First edition 2011-12-01

Systems and software engineering — Life cycle processes — Requirements engineering

Ingénierie des systèmes et du logiciel — Processus du cycle de vie — Ingénierie des exigences

At feasibility & Early Requirements



What other software functions are not visible yet?

Software Initial Requirements – ISO 29148

Contextual

1. Purpose
2. Scope
3. Product perspective
4. Product functions
5. User characteristics
6. Limitations
7. Assumptions & dependencies
8. Apportioning of requirements
9. Specified requirements

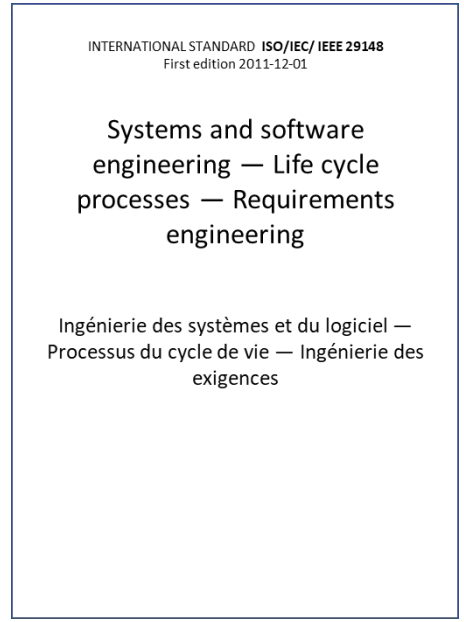
1. Verification
2. Supporting documents

External Interfaces

Functions

Non-Functional & Quality

1. Usability requirements
2. Performance requirements
3. Logical database requirements
4. Design constraints
5. Standards compliance
6. Software system attributes


INTERNATIONAL STANDARD **ISO/IEC/ IEEE 29148**
First edition 2011-12-01

Systems and software engineering — Life cycle processes — Requirements engineering

Ingénierie des systèmes et du logiciel — Processus du cycle de vie — Ingénierie des exigences

Software Lifecycle in practice: Early coding

Contextual

1. Purpose
2. Scope
3. Product perspective
4. Product functions
5. User characteristics
6. Limitations
7. Assumptions & dependencies
8. Apportioning of requirements
9. Specified requirements

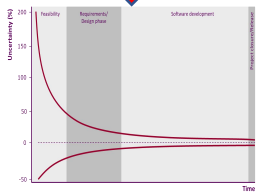
1. Verification
2. Supporting doc.

External Interfaces

Functions

Non-Functional & Quality

1. Usability req.
2. Performance req.
3. Logical database req.
4. Design constraints
5. Standards compliance
6. Software system attributes



Detailed Software Functions

What Size is still NOT VISIBLE?

Functionality Specified & Approved

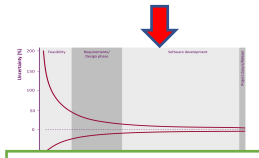
What other software functions are not visible yet?

Late at Initial Testing

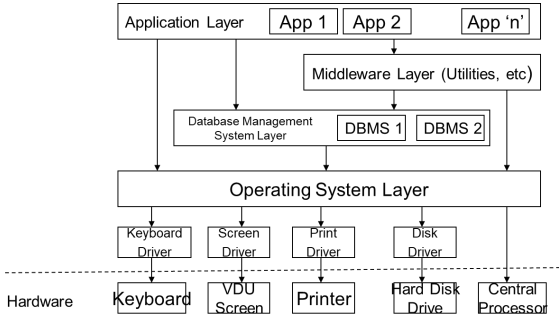
Contextual

1. Purpose
2. Scope
3. Product perspective
4. Product functions
5. User characteristics
6. Limitations
7. Assumptions & dependencies
8. Apportioning of requirements
9. Specified requirements

External Interfaces

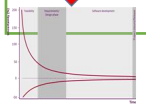


Software Functions



Functions

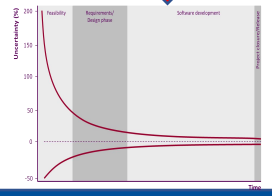
Detailed Software Functions



Non-Functional & Quality

1. Usability req.
2. Performance req.
3. Logical database req.
4. Design constraints
5. Standards compliance
6. Software system attributes

Software Functions



1. Verification
2. Supporting doc.

Very late at Testing & Implementation!

Contextual

1. Purpose
2. Scope
3. Product perspective
4. Product functions
5. User characteristics
6. Limitations
7. Assumptions & dependencies
8. Apportioning of requirements
9. Specified requirements

1. Verification
2. Supporting doc.

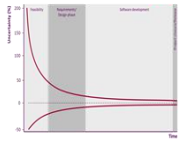
External Interfaces → **Software Functions** → **Detailed Software Functions**

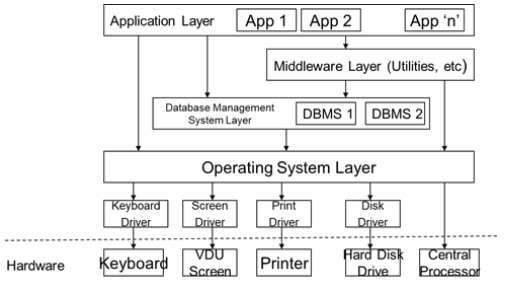
Functions → **Detailed Software Functions**

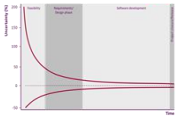
Non-Functional & Quality

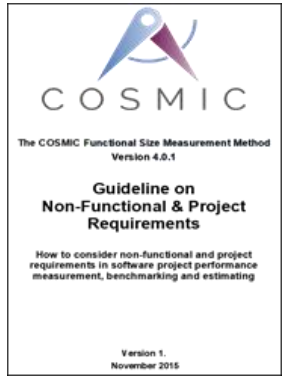
1. Usability req.
2. Performance req.
3. Logical database req.
4. Design constraints
5. Standards compliance
6. Software system attributes

→ **Software Functions** → **Detailed Software Functions**

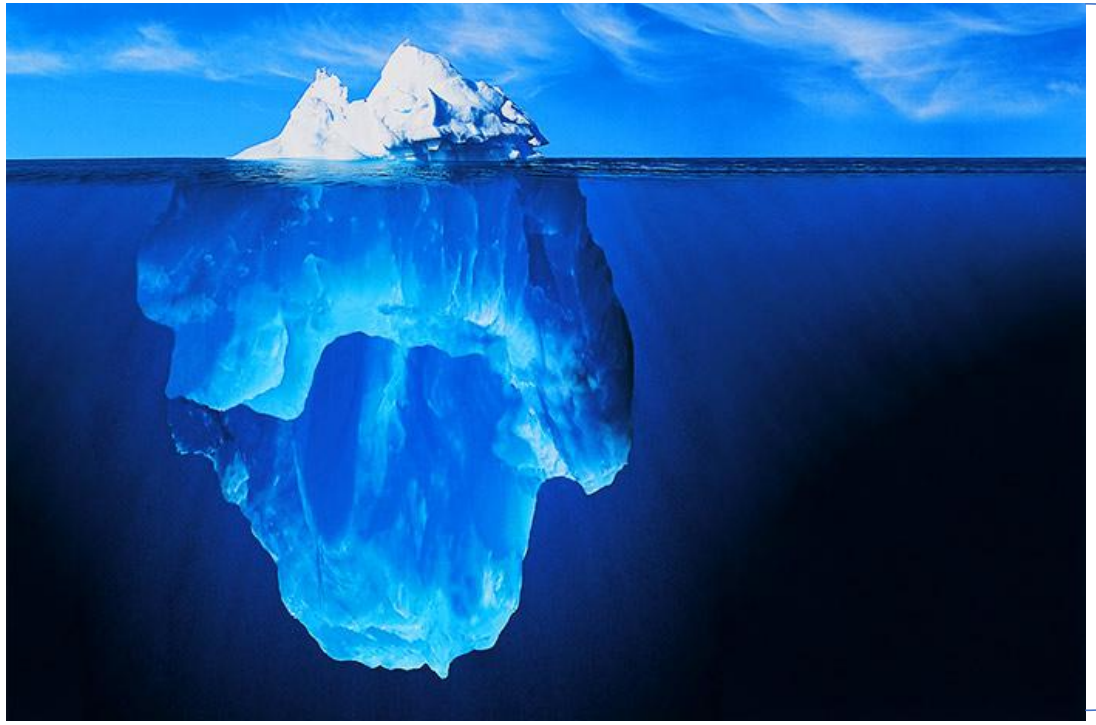






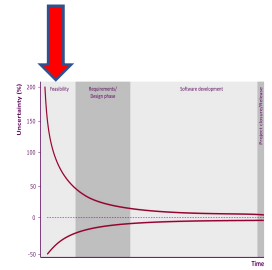


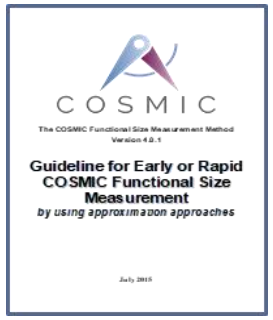
What is VISIBLE at Project Completion?

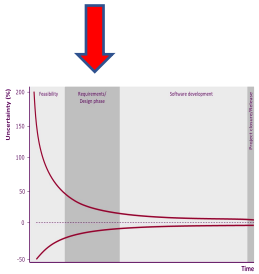


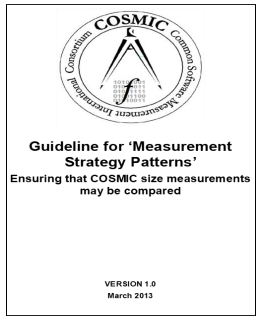
All software functions available for accurate measurement

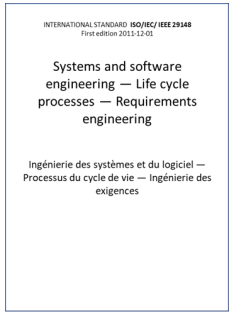
COSMIC Guidelines for Estimation through the lifecycle

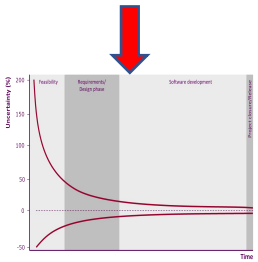


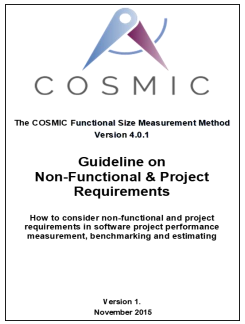


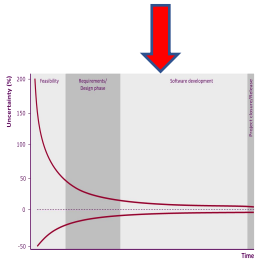


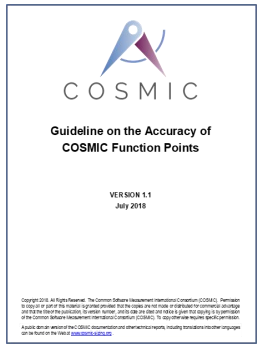


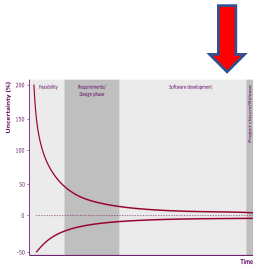




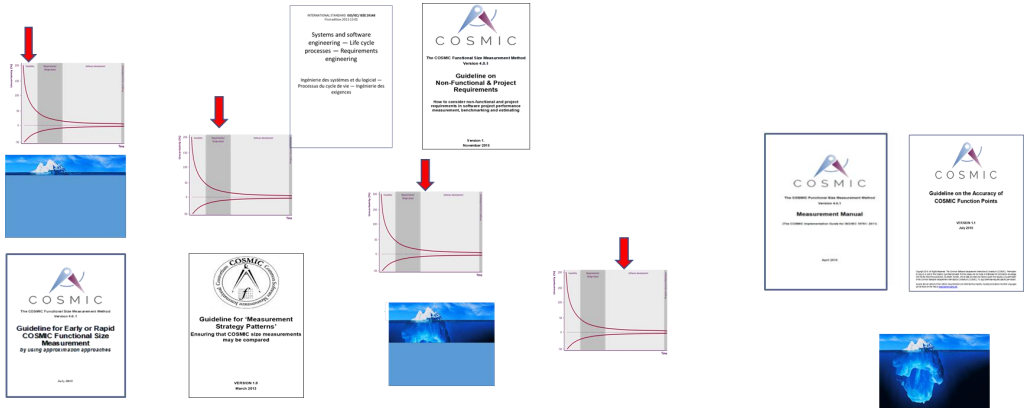






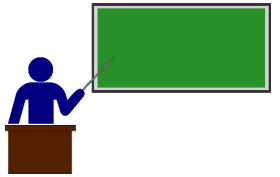


What's next?



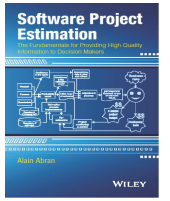
New COSMIC Tutorials at this conference

- Accuracy of Measurements
- COSMIC Size Estimation



Tutorials available

- Effort Estimation Models



Upcoming **Certification**

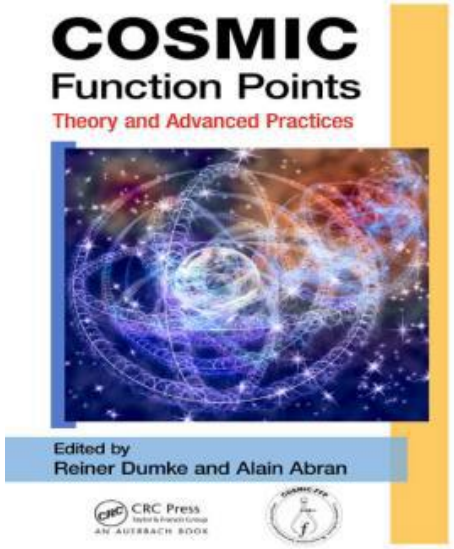
- COSMIC **Size** Estimation
- Effort Estimation Models

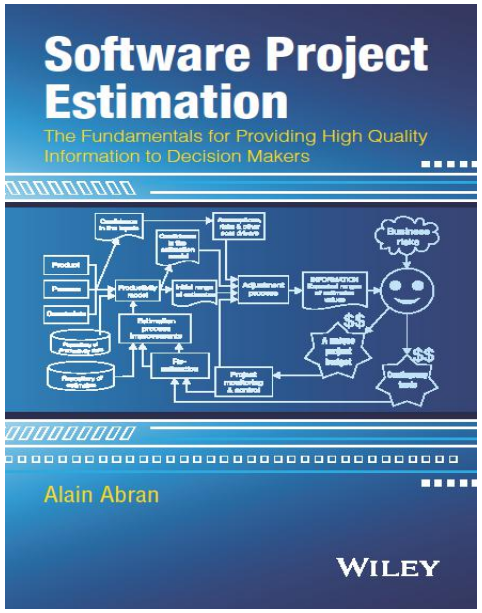
Questions?

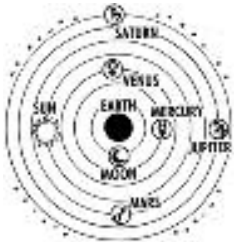

THANK YOU

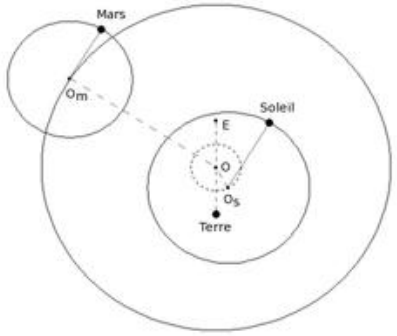
Other sources of COSMIC examples with industry data

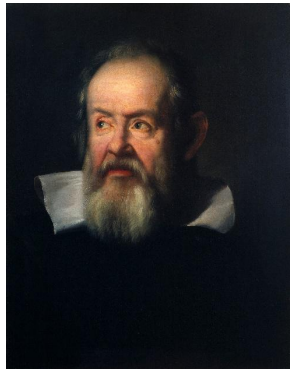
- COSMIC web site at: www.cosmic-sizing.org

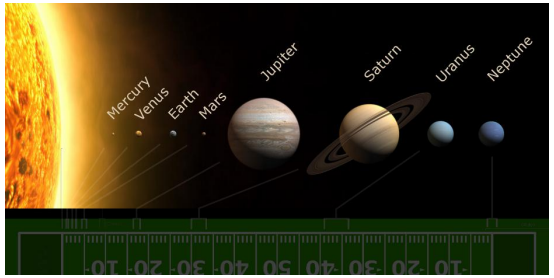


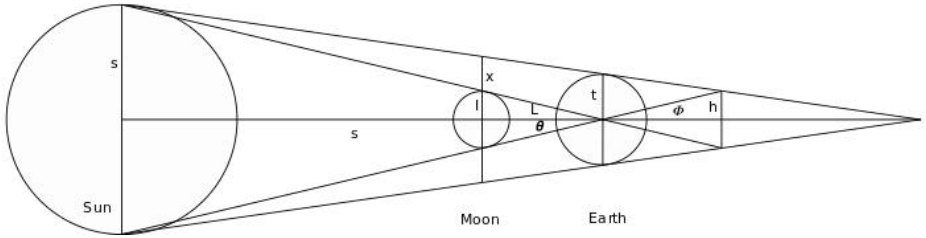


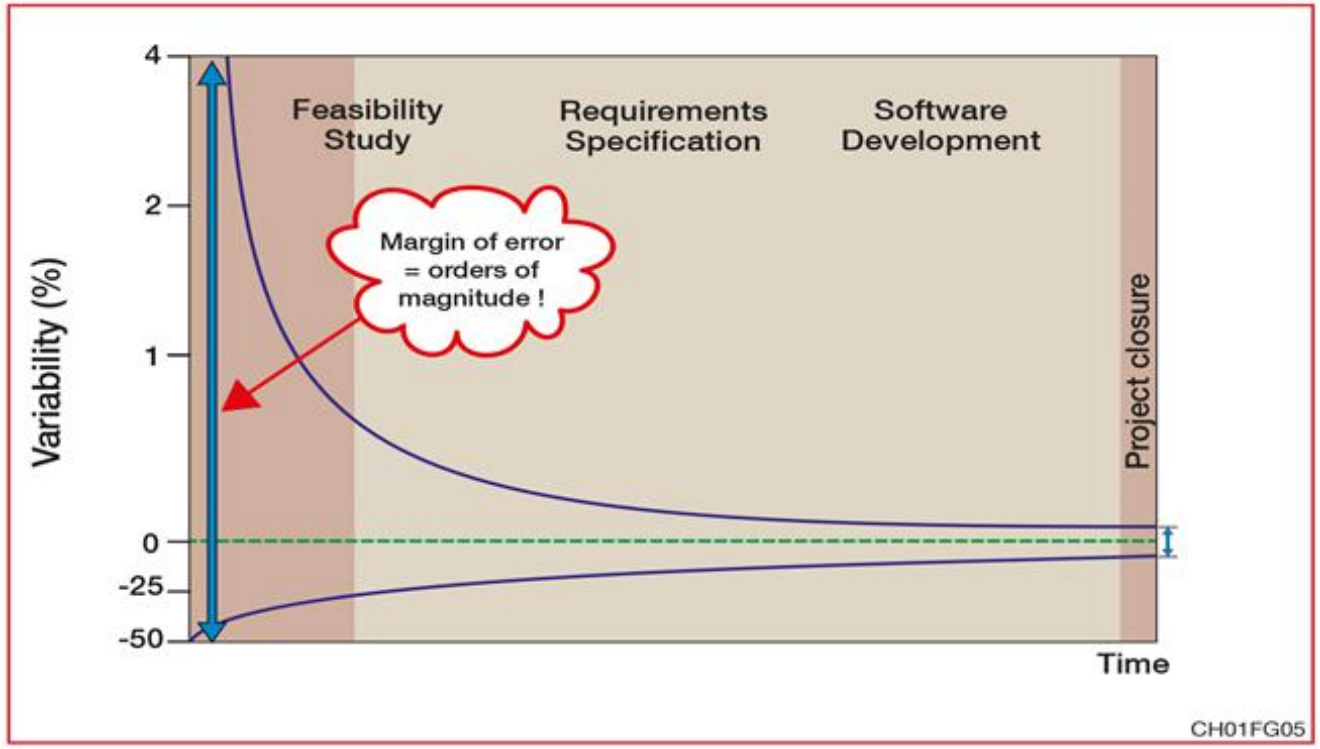


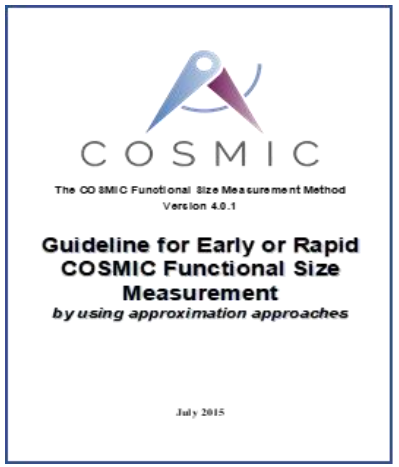


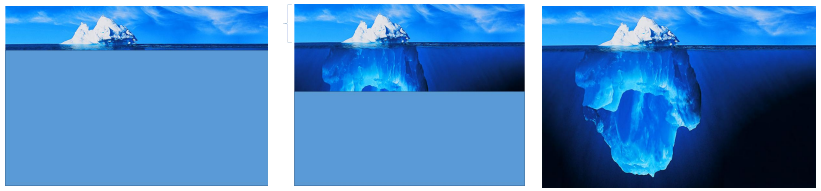


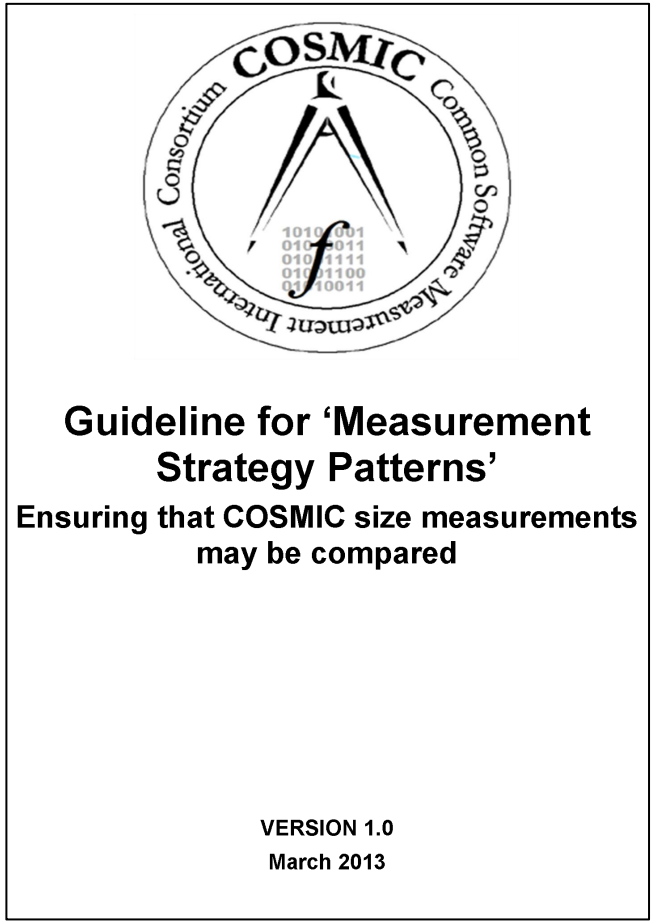


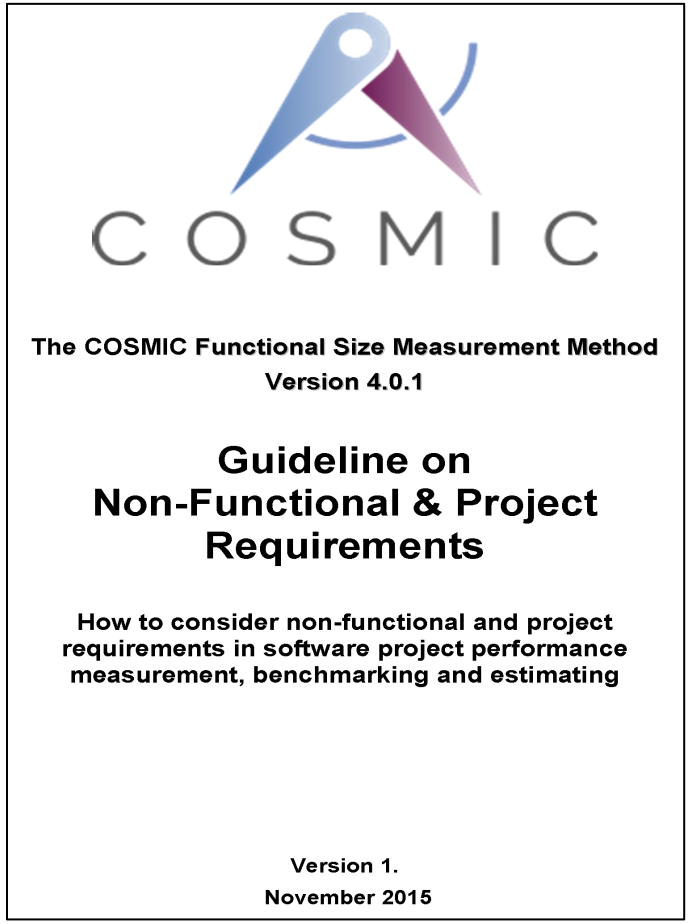












In practice: Very late at Testing & Implementation!

Contextual

1. Purpose
2. Scope
3. Product perspective
4. Product functions
5. User characteristics
6. Limitations
7. Assumptions & dependencies
8. Apportioning of requirements
9. Specified requirements

1. Verification
2. Supporting doc.

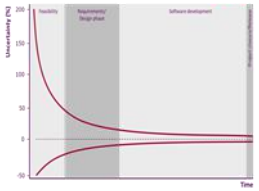
External Interfaces → **Software Functions** → **Detailed Software Functions**

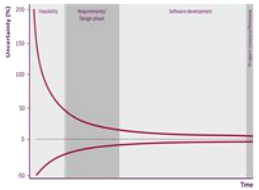
Functions → **Detailed Software Functions**

Non-Functional & Quality

1. Usability req.
2. Performance req.
3. Logical database req.
4. Design constraints
5. Standards compliance
6. Software system attributes

→ **Software Functions** → **Detailed Software Functions**





Phases:
A: Estimation of the Inputs for estimation…
B: Execution of the productivity model