National Congress of Measurement and Software Estimation

Booklets

RENIECYT
Registro Nacional de Instituciones y Empresas Científicas y Tecnológicas
1702902 CONACYT

RENIECYT - LATINDEX - Research Gate - DULCINEA - CLASE - Sudoc - HISPANA - SHERPA UNIVERSIA - Google Scholar DOI - REDIB - Mendeley - DIALNET - ROAD - ORCID

Title: Early Sizing of Software: 'Finger in the air' or 'Engineering techniques'?

Author: Alain Abran

Editorial label ECORFAN: 607-8695
BCNMES Control Number: 2020-02
BCNMES Classification (2020): 110820-0002

Pages: 62
RNA: 03-2010-032610115700-14

ECORFAN-México, S.C.
143 – 50 Itzopan Street
La Florida, Ecatepec Municipality
Mexico State, 55120 Zipcode
Phone: +52 1 55 6159 2296
Skype: ecorfan-mexico.s.c.
E-mail: contacto@ecorfan.org
Facebook: ECORFAN-México S. C.
Twitter: @EcorfanC

www.ecorfan.org

Holdings

Mexico
Bolivia
Spain
Ecuador
Peru

Colombia
Cameroon
El Salvador
Taiwan
Paraguay

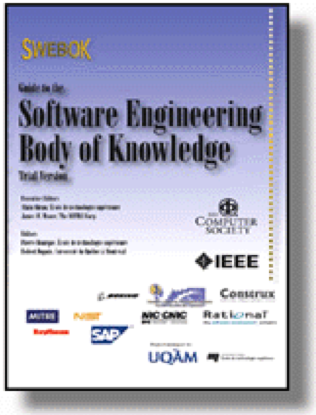
Guatemala
Democratic
Republic
of Congo
Nicaragua

Alain Abran

20 years

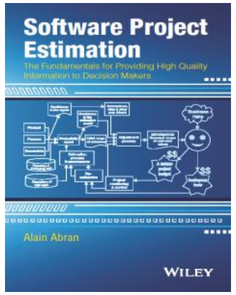
- **Development**
- **Maintenance**
- **Process Improvement**

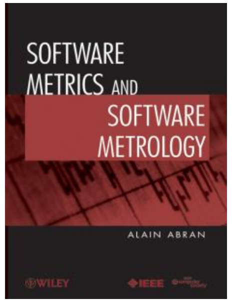
+ 20 years


SWEBOK
Guide to the Software Engineering Body of Knowledge
Trial Version

45 PhD

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


Software Project Estimation
The Fundamentals for Providing High Quality Information to Decision Makers
Alain Abran
Wiley


Software Metrics and Software Metrology
Alain Abran
Wiley IEEE

Software Maintenance Management
Evaluation and Continuous Improvement
Alain April Alain Abran

List of topics

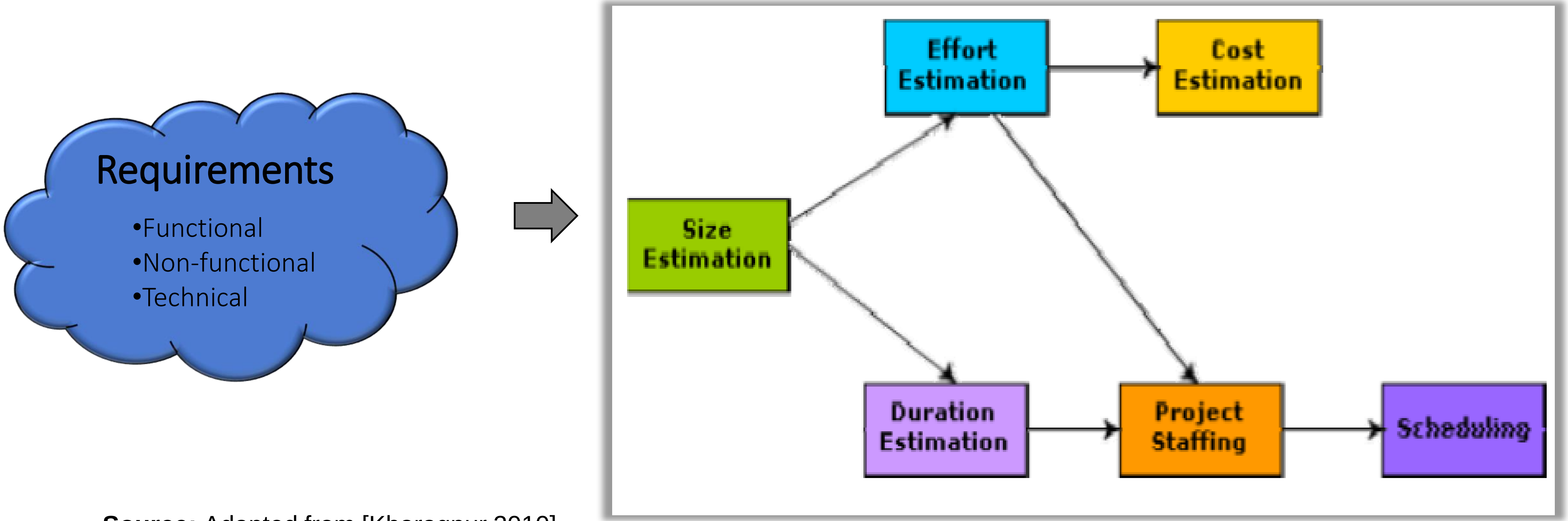
1. Early Size Estimation issues

2. Size approximation techniques

3. The software-iceberg analoly for early sizing

4. Wrap-up

The general estimation process



Source: Adapted from [Kharagpur 2010]

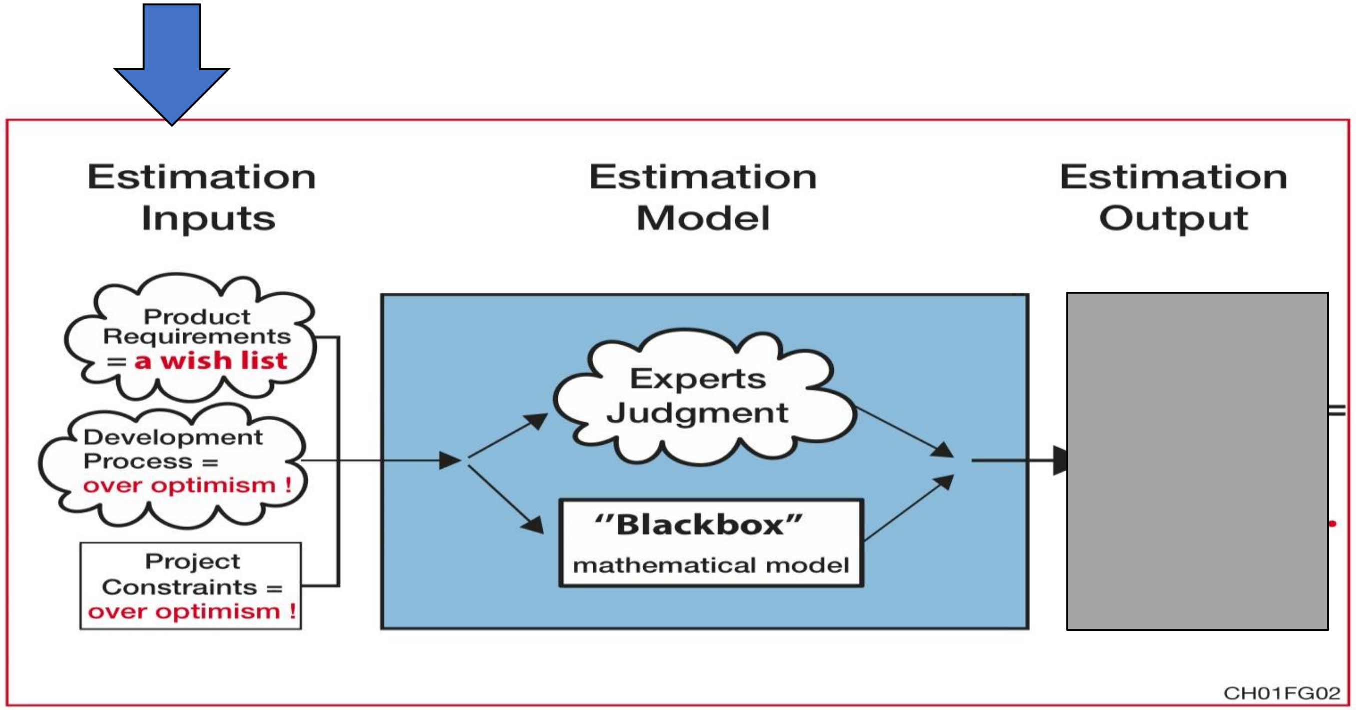
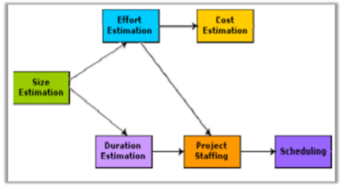
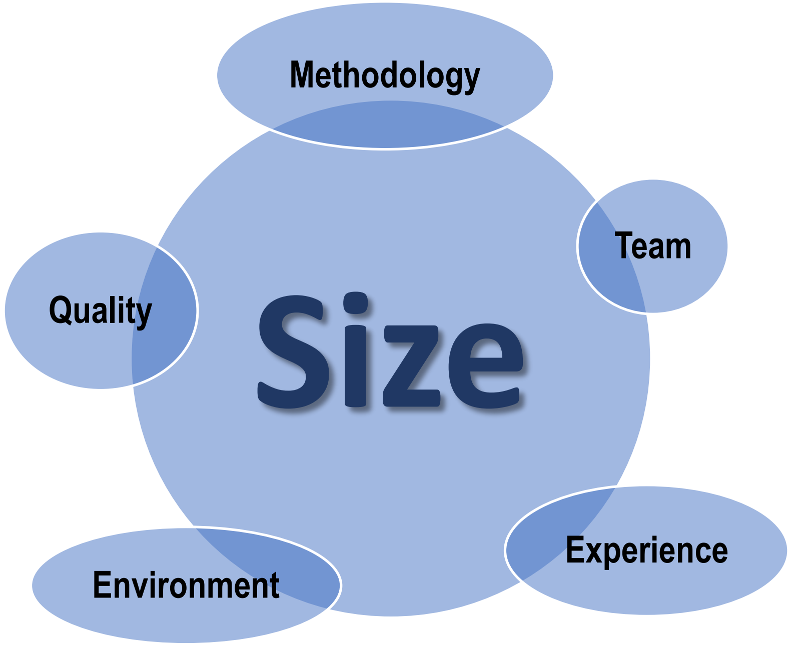

Estimation Inputs
Estimation Model
Estimation Output
Product Requirements = a wish list
Development Process = over optimism !
Project Constraints = over optimism !
Experts Judgment
"Blackbox" mathematical model
CH01FG02

Figure 1.2 Some poor estimation practices observed in industry.

Size: Dominant variable





Methodology

Team

Quality

Size

Experience

Environment

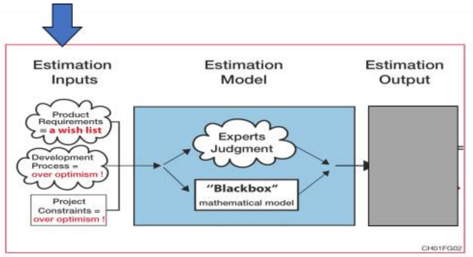
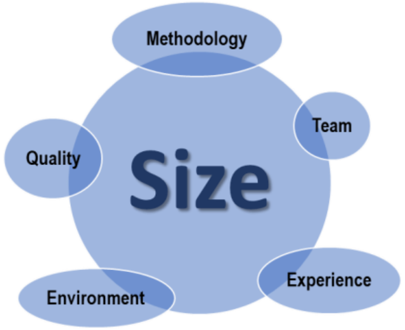
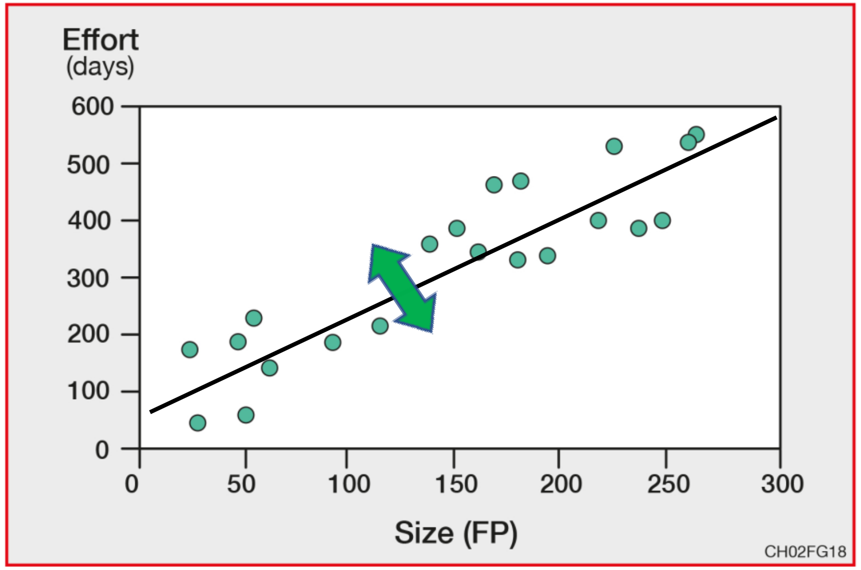

Estimation Inputs
Estimation Model
Estimation Output
Product Requirements = a wish list
Development Process = over optimism !
Project Constraints = over optimism !
Experts Judgment
"Blackbox" mathematical model
CH01FG02

Figure 1.2 Some poor estimation practices observed in industry.

Size as a dominant variable in a dataset



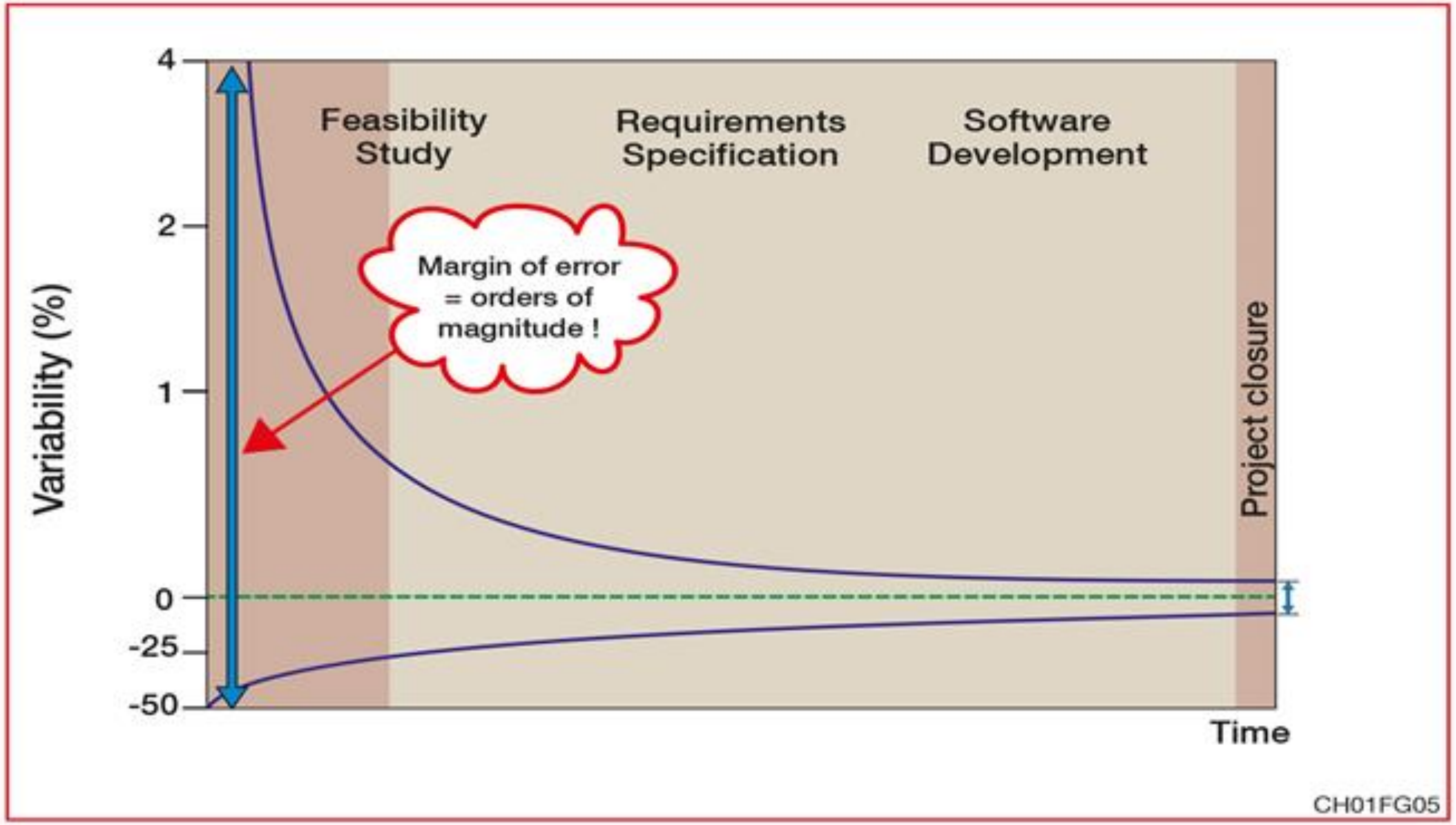


Homogeneous dataset of 21 projects (Abran 1994)

In a development process under control:

- Size explains **80 to 90%** of the Effort variation

- **all other factors combined impact 5% to 20%!**


4
2
1
0
-25
-50
Variability (%)
Feasibility Study
Requirements Specification
Software Development
Project closure
Margin of error = orders of magnitude !
Time
CH01FG05

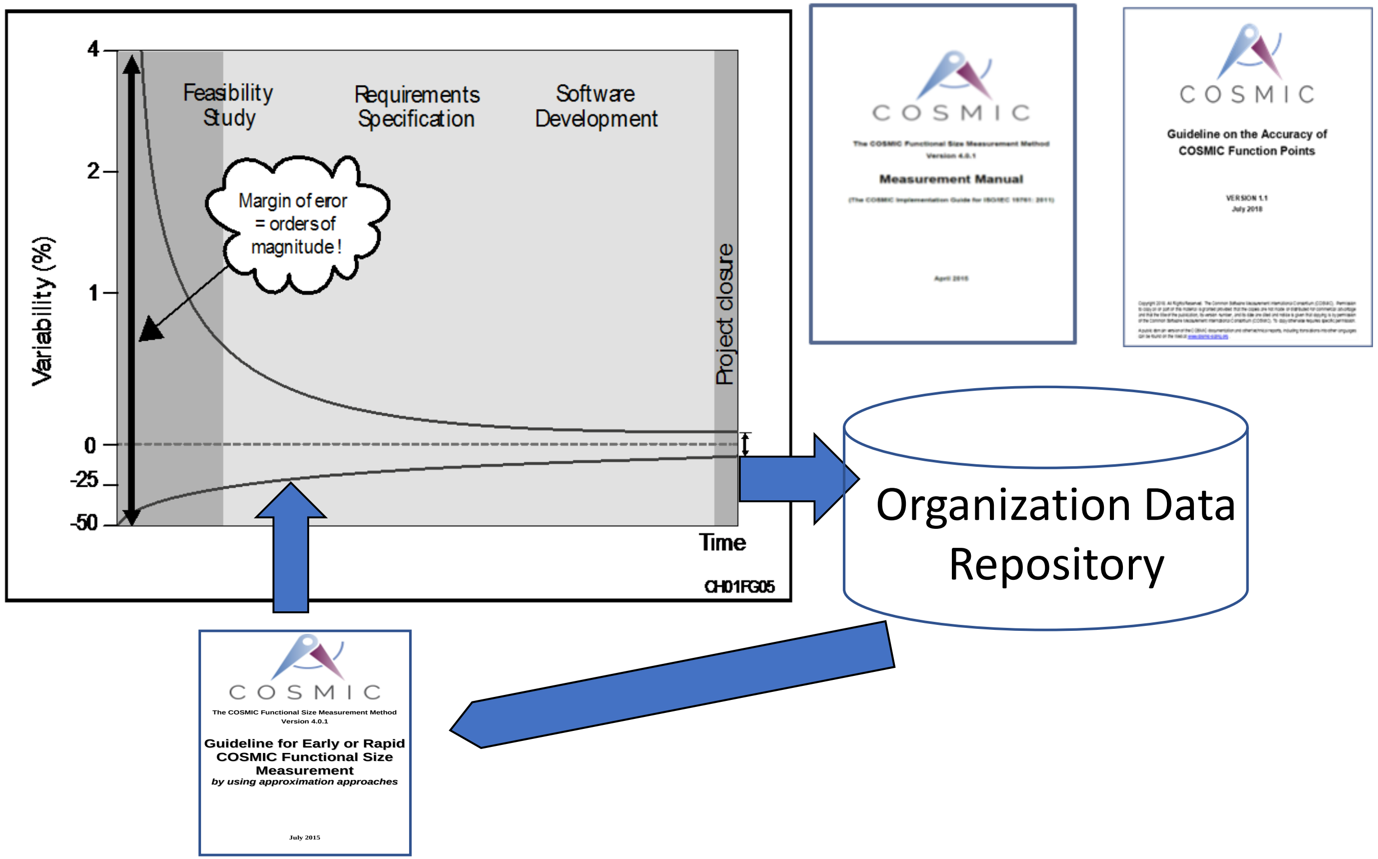
Measurement vs Estimation of Size

Accurate measurement:

- ➢ end of development

Estimation of size:

- ➢ early phases


4
2
1
0
-25
-50
Variability (%)
Feasibility Study
Requirements Specification
Software Development
Project closure
Margin of eror = orders of magnitude !
Time
CH01FG05
Organization Data Repository

COSMIC

The COSMIC Functional Size Measurement Method
Version 4.0.1

Measurement Manual

(The COSMIC Implementation Guide for ISO/IEC 19761: 2011)

April 2015

COSMIC

Guideline on the Accuracy of COSMIC Function Points

VERSION 1.1
July 2018

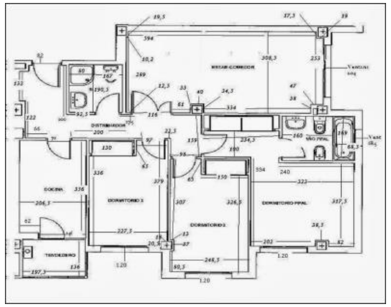
COSMIC

The COSMIC Functional Size Measurement Method
Version 4.0.1

Guideline for Early or Rapid COSMIC Functional Size Measurement
by using approximation approaches

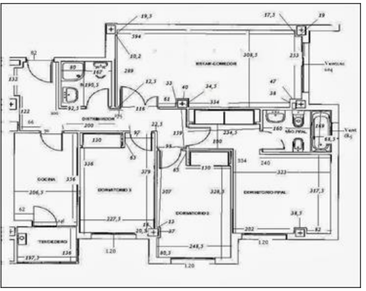
July 2015

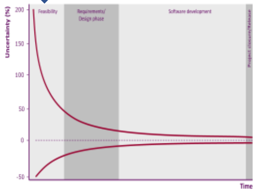
Engineering

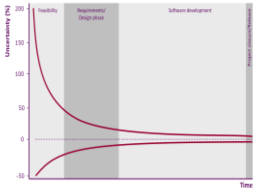


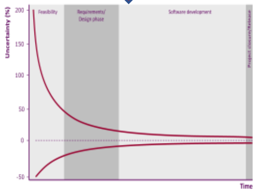
- ✓Detailed observation of past projects
- ✓Quantitative data collection
- ✓Analysis of the impact of individual variables (one at a time)
- ✓Selection of relevant samples
- ✓Statistical analyses
- ✓Very careful extrapolation to similar or other contexts

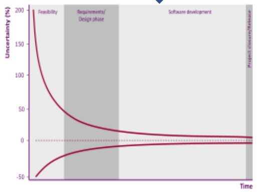
Size & Estimation throughout the lifecycle

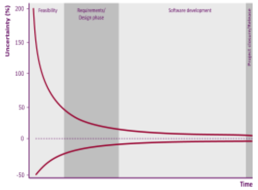








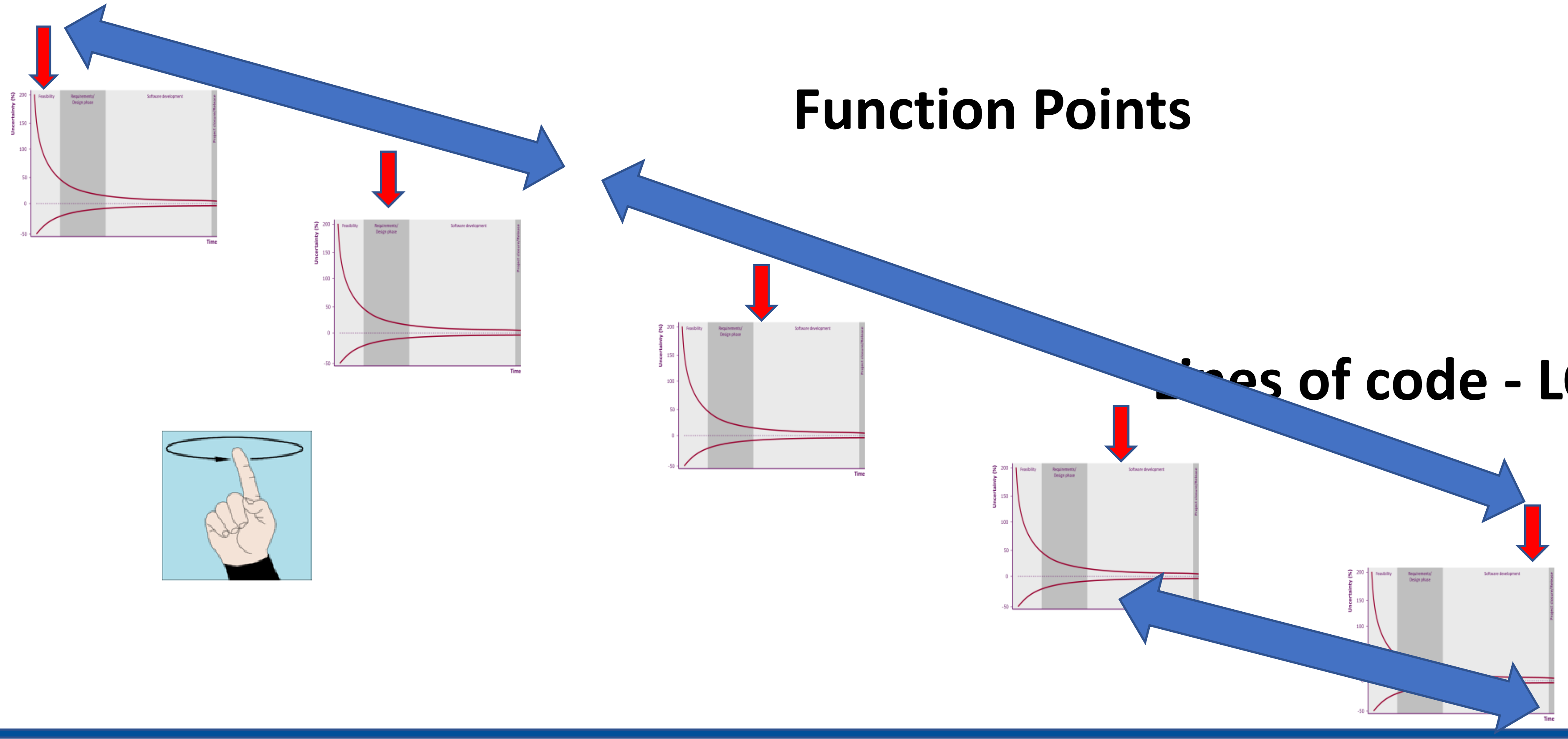




List of topics

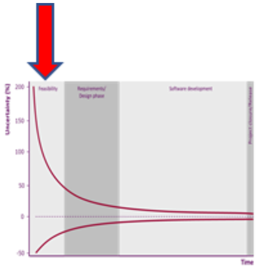
1. Early Size Estimation issues
2. Size approximation techniques
3. The software-iceberg analoly for early sizing
4. Wrap-up

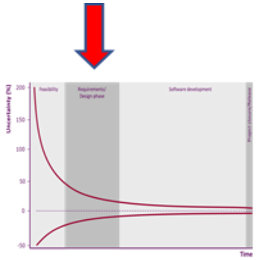
Size Techniques throughout the lifecycle



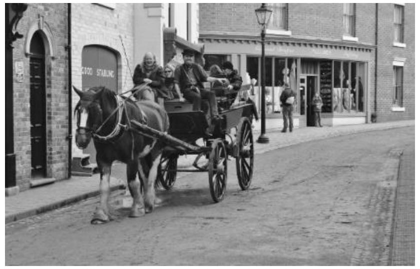
Function Points

Lines of code - LOC





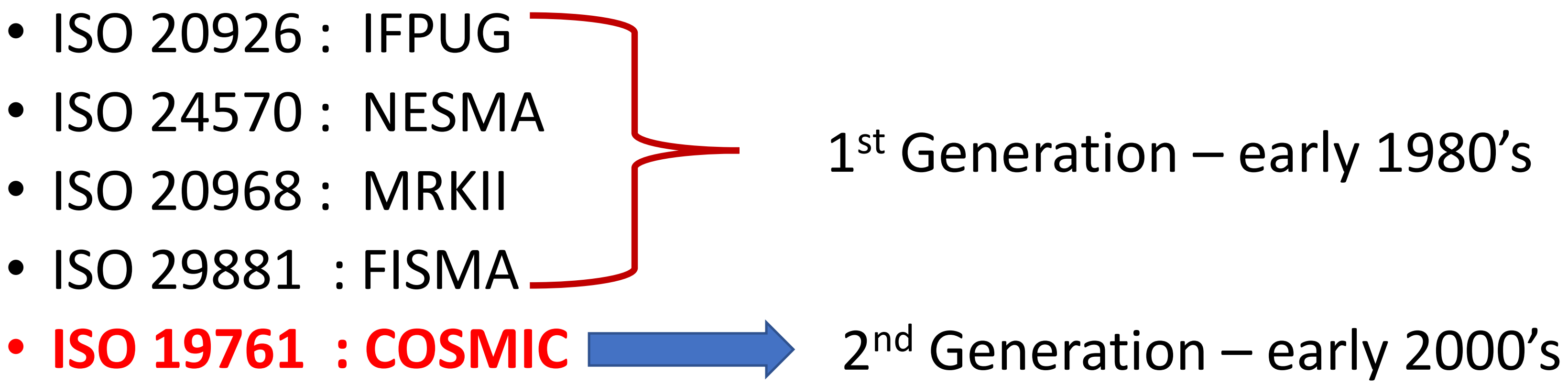
Lines of code-based estimation models



Function Points-based estimation models

Function Points

5 distinct ISO standards



- ISO 20926 : IFPUG
- ISO 24570 : NESMA
- ISO 20968 : MRKII
- ISO 29881 : FISMA
- **ISO 19761 : COSMIC**

1st Generation – early 1980’s

2nd Generation – early 2000’s

COSMIC Function Points - ISO 19761

Every software is different, but …..

what is common across all software:

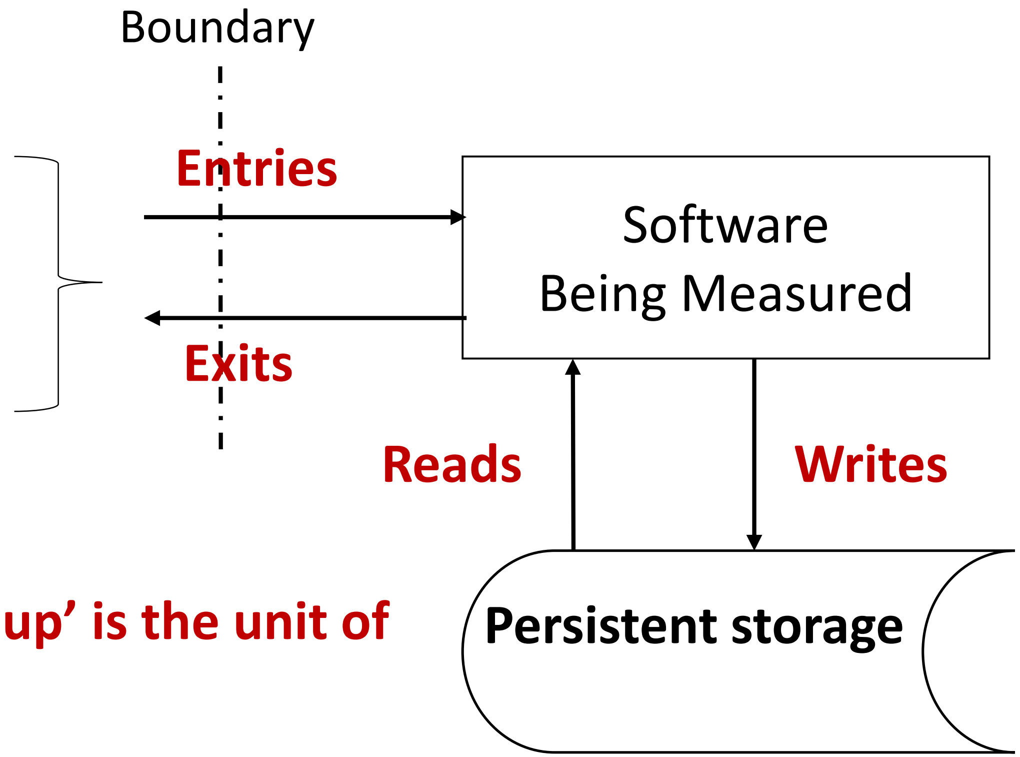
- In different types of software?
- In very small software?
- In very large software?
- In distinct software domains?
- In various countries?

COSMIC view of software COSMIC

All software does this:

Functional Users types:

1. Humans
2. Hardware devices
3. Other software



Boundary

Entries

Exits

Software
Being Measured

Reads

Writes

Persistent storage

The ‘Data Movement of 1 data group’ is the unit of measurement: 1 CFP
(1 CFP = 1 COSMIC Function Point)

COSMIC: Applicable to all software domains

COSMIC Guidelines:

1. Business applications
2. Real-time software
3. Data Warehouse software
4. SOA software (SOA: Service Oriented Architecture)
5. Mobile apps
6. Agile Development

COSMIC

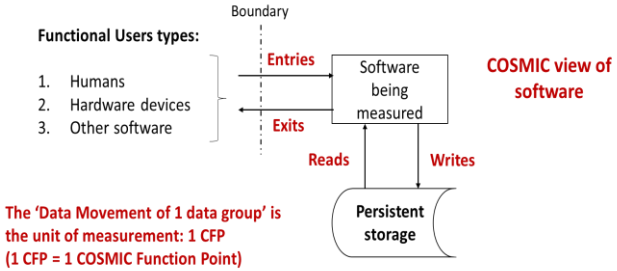
The COSMIC Functional Size Measurement Method
Version 4.0.1

Guideline for Sizing
Business Application Software

VERSION 1.3a
Febuary 2016

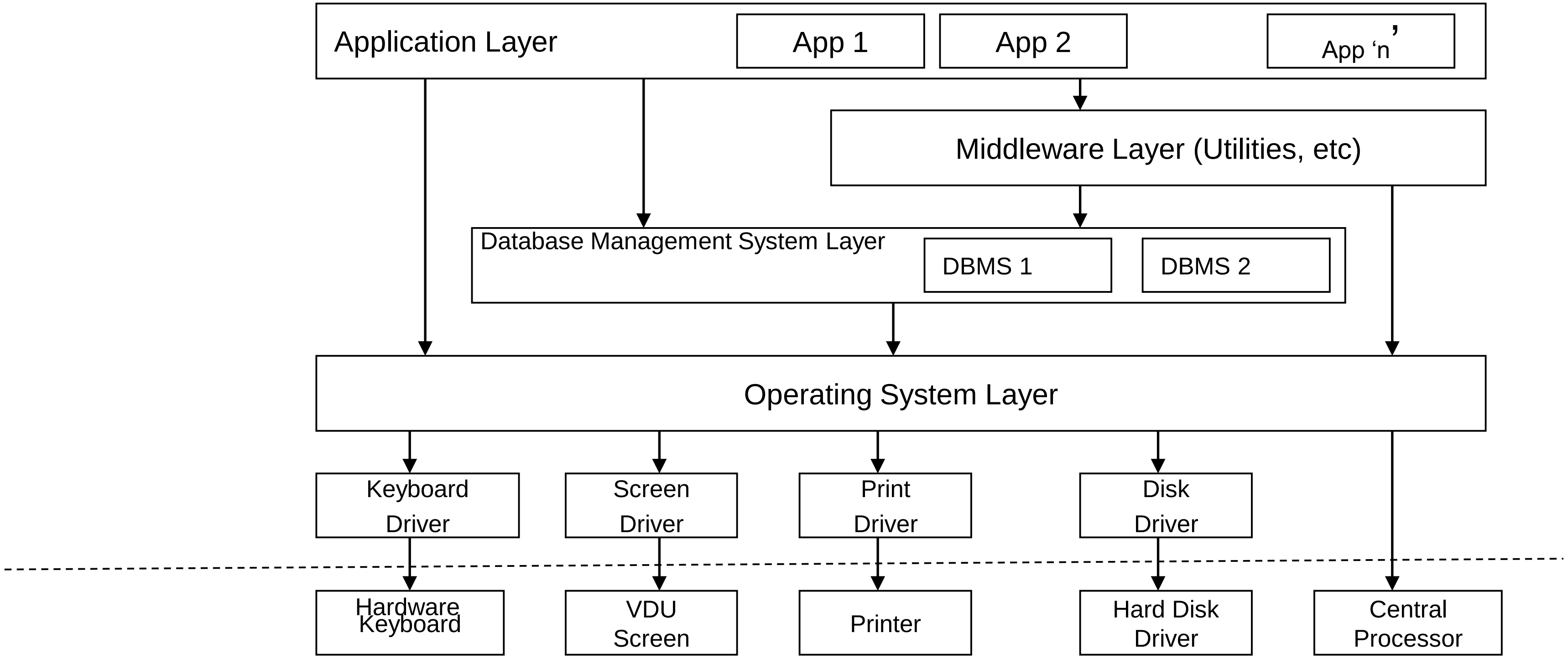
All COSMIC documents free on the web at: www.cosmic-sizing.org

COSMIC – applicable throughout a software architecture

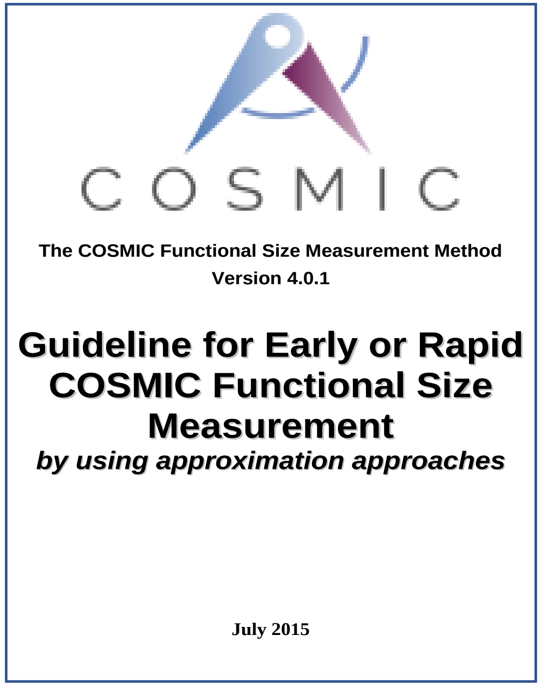


COSMIC view of software

The 'Data Movement of 1 data group' is the unit of measurement: 1 CFP
(1 CFP = 1 COSMIC Function Point)



COSMIC at Estimation Time



COSMIC

The COSMIC Functional Size Measurement Method
Version 4.0.1

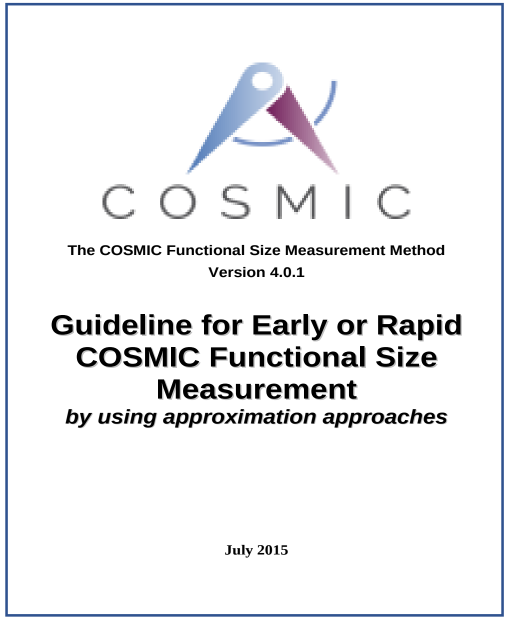
Guideline for Early or Rapid COSMIC Functional Size Measurement
by using approximation approaches

July 2015

Approximation techniques (with reported use, strengths & weaknesses)

1. Average functional process
2. Fixed size classification
3. Equal size bands
4. Average use case
5. Easy function points
6. Approx. from informally written texts
7. EPCU – Fuzzy logic
8. Etc.

Approximation techniques



COSMIC

The COSMIC Functional Size Measurement Method
Version 4.0.1

Guideline for Early or Rapid COSMIC Functional Size Measurement
by using approximation approaches

July 2015

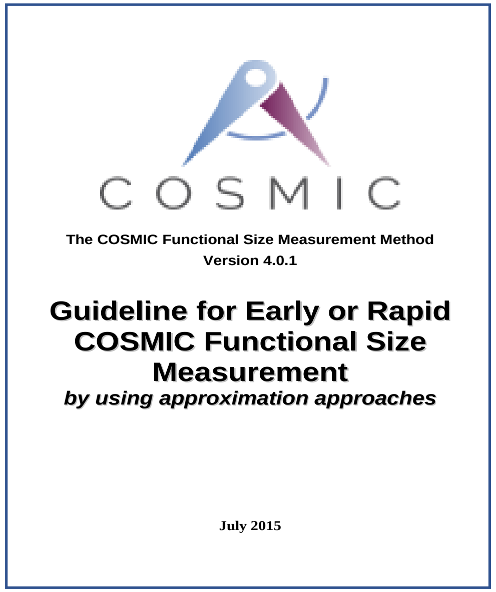
❖Requirements stage:

1. Average functional process
2. Average use case
3. Fixed size band
4. Equal size band
5. Functional Patterns

❖Feasibility stage:

1. Early & Quick sizing
2. Easy sizing
3. EPCU-Fuzzy logic
4. ……

Approximation techniques



COSMIC

The COSMIC Functional Size Measurement Method
Version 4.0.1

Guideline for Early or Rapid COSMIC Functional Size Measurement
by using approximation approaches

July 2015

❖Requirements stage:

1. Average functional process
2. Average Use case
3. Fixed size band
4. Equal size band
5. Functional Patterns

❖Feasibility stage:

1. Early & Quick sizing
2. Easy sizing
3. EPCU-Fuzzy logic
4. ……

Average Functional Process Approximation

Steps for Sampling & Calculation of an **average functional process**:

1. **From past projects, identify a sample** of requirements whose functional processes & data movements have been defined in detail.
2. **Identify the functional processes** of this sample.
3. **Measure precisely** the sizes of the functional processes of the sample.
4. **Calculate the average size**, in CFP, of the functional processes in the sample
   - average size = 8 CFP per Functional Process

- **-> '8' is the scaling factor**

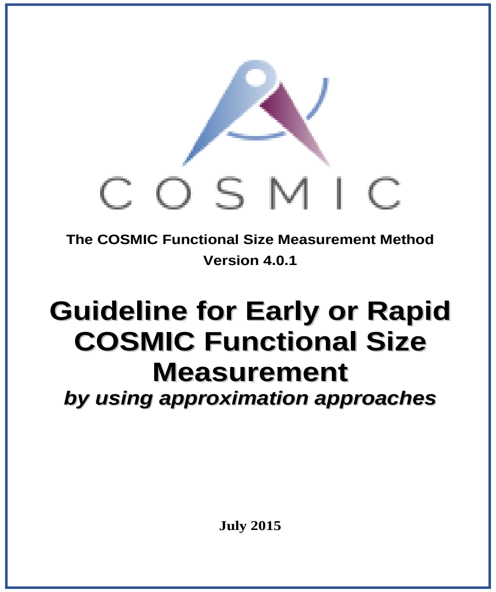
Average Functional Process Approximation

Approximation **using** the average of the sample

1. Identify & count all functional processes for which the size has to be estimated early:
   - ➢ 40 Functional Processes
2. Estimated functional size =
   - ➢ Number of functional processes x scaling factor
     = 40 x 8 = 320 CFP

✓ **Valid as long as the sample used to calculate the size of the average functional process is representative for the software being estimated.**

Approximation techniques



COSMIC

The COSMIC Functional Size Measurement Method
Version 4.0.1

Guideline for Early or Rapid COSMIC Functional Size Measurement
by using approximation approaches

July 2015

❖Requirements stage:

1. Average functional process
2. Average Use case
3. Fixed size band
4. Equal size band
5. Functional Patterns

❖Feasibility stage:

1. Early & Quick sizing
2. Easy sizing
3. EPCU-Fuzzy logic
4. ……

Average Use Case Approximation

26 Example: A Use case with:

- 6 functional processes on average for a use case
- each functional process on average size = 8 CFP [range + or – 1 Std CFP]
  - Hence the average size of a use case = 8 x 6 = **48 CFP per use case**.

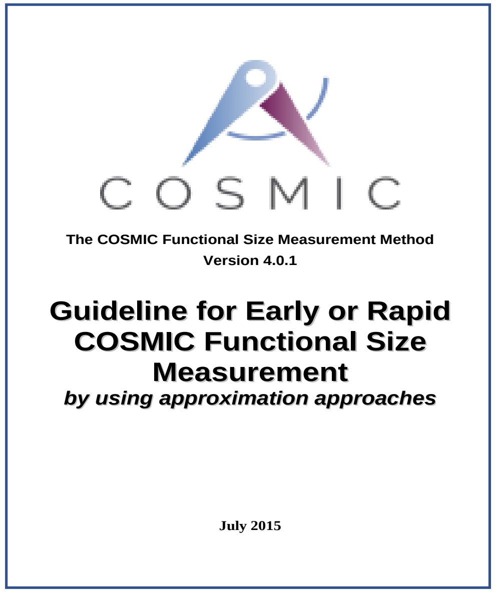
For a new project with 12 use cases **= ?**

- software size would be 12 x 48 = 576 CFP [range + or – 1 Std CFP]

Note: The uncertainty on this approximate size will be greater:

- the scale factor 48 is the product of 2 scale factors (8 & 6) which are themselves estimated.

Approximation techniques



COSMIC

The COSMIC Functional Size Measurement Method
Version 4.0.1

Guideline for Early or Rapid COSMIC Functional Size Measurement
by using approximation approaches

July 2015

- ❖Requirements stage:
  1. Average functional process
  2. Average Use case
  3. Fixed size band
  4. Equal size band
  5. Functional Patterns
- ❖Feasibility stage:
  1. Early & Quick sizing
  2. Easy sizing
  3. EPCU-Fuzzy logic
  4. ……

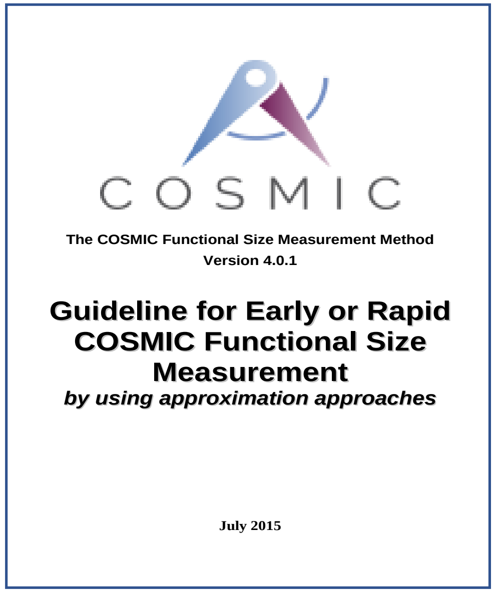
Fixed Size Classification Approximation

Example:

3 size classes with sizes based on an expected number of data movements

Classification	Size (CFP)	#E	#X	#R	#W	Error messages
Small	5	1	1	1	1	1
Medium	10	2	2	3	2	1
Large	15	3	3	4	4	1
…						

Approximation techniques



COSMIC

The COSMIC Functional Size Measurement Method
Version 4.0.1

Guideline for Early or Rapid COSMIC Functional Size Measurement
by using approximation approaches

July 2015

❖Requirements stage:

1. Average functional process
2. Average Use case
3. Fixed size band
4. Equal size band
5. Functional Patterns

❖Feasibility stage:

1. Early & Quick sizing
2. Easy sizing
3. EPCU-Fuzzy logic
4. ……

Equal Size Bands Approximation

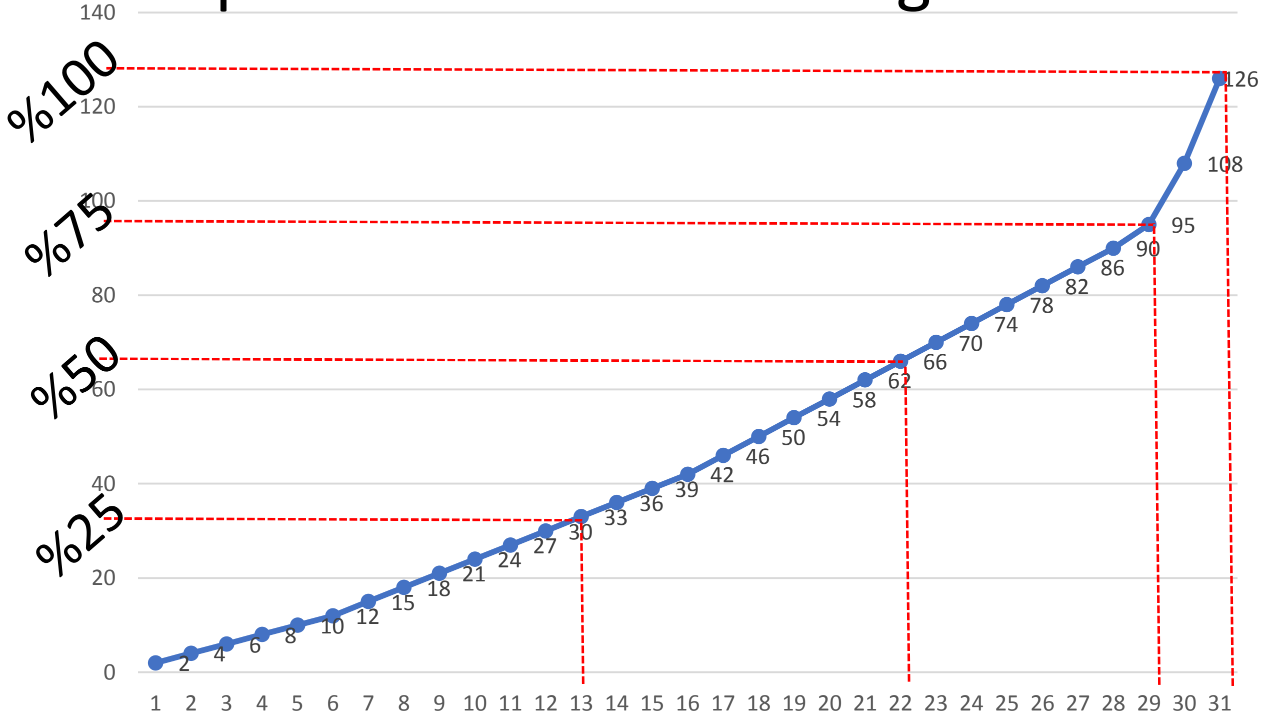
Equal size bands boundaries are chosen so that the **total size of all the functional processes in each band** is the same for each band.

Examples:

- If 3 bands are used:
  - ➢ total size of all functional processes in each band = 33% of total size.
- If 5 bands are used:
  - ➢ total size of all functional processes in each band = 20% of total size.

Equal Size Band EXAMPLE

COSMIC Restaurant case study with data from 31 functional processes -in increasing size



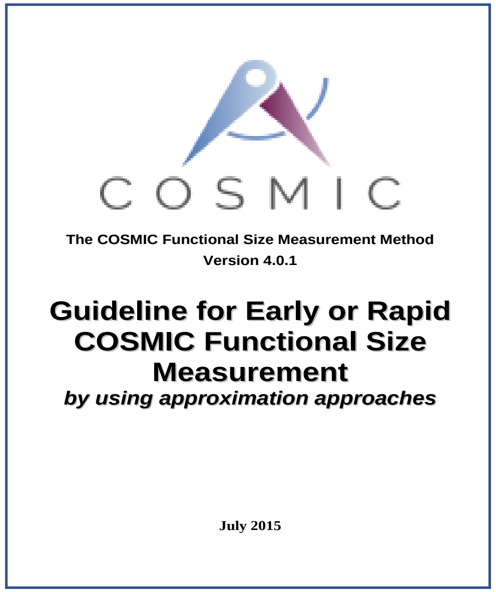
Average sizes per band.

- $\frac{30}{13} = 2.3\ \mathtt{CFP}$
- $\frac{(62-30)}{(22-13)} = \frac{28}{9} = 3.1\ \mathtt{CFP}$
- $\frac{(95-62)}{(29-22)} = \frac{33}{7} = 4.7\ \mathtt{CFP}$
- $\frac{(126-95)}{(31-29)} = \frac{31}{2} = 15.5\ \mathtt{CFP}$

Equal Size Band EXAMPLE – Resto-Sys

Band	Average size of a Functional Process	% of total Functional Size	% of total number of Functional Processes
Small	2.3	25%	42%
Medium	3.1	25%	29%
Large	4.7	25%	23%
Very Large	15.5	25%	6%

Approximation techniques



COSMIC

The COSMIC Functional Size Measurement Method
Version 4.0.1

Guideline for Early or Rapid COSMIC Functional Size Measurement
by using approximation approaches

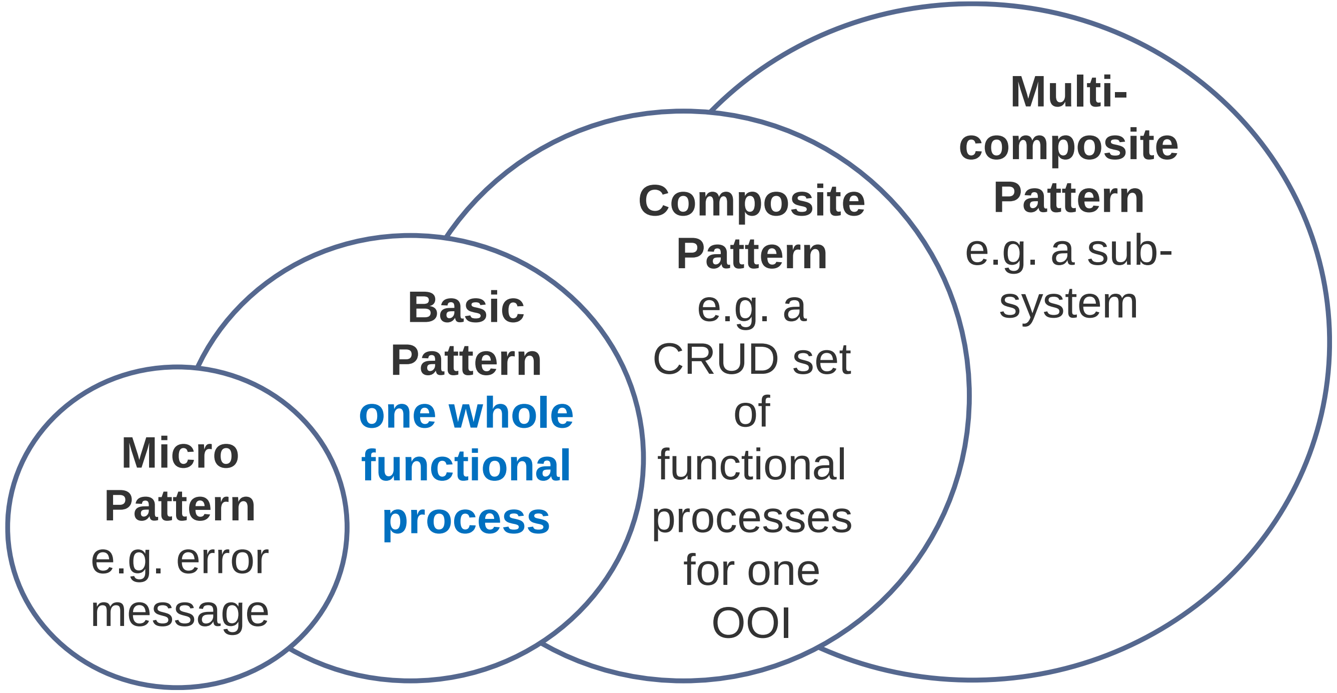
July 2015

- Requirements stage:
  1. Average functional process
  2. Average Use case
  3. Fixed size band
  4. Equal size band
  5. Functional Patterns
- Feasibility stage:
  1. Early & Quick sizing
  2. Easy sizing
  3. EPCU-Fuzzy logic
  4. ……

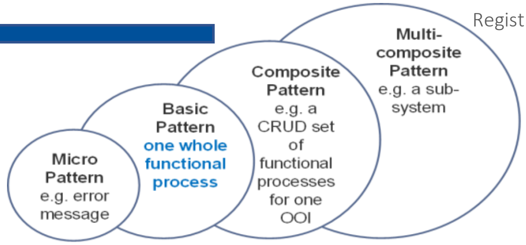
Functional Size Measurement Patterns

34 Observations of measurers:

- some patterns of measurement results recur repeatedly.
- Four types of patterns



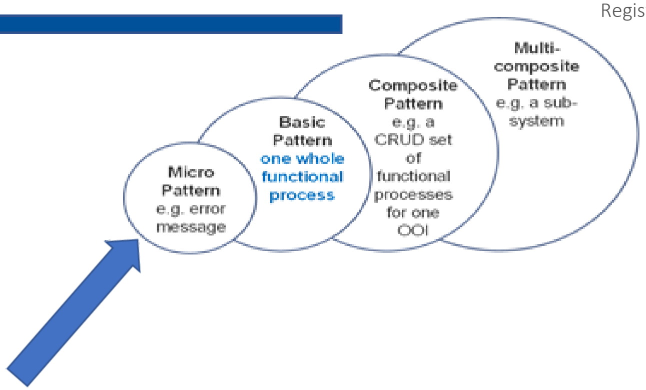
Functional Size Patterns



35

- **Micro FSM patterns:** A fragment of a functional process, involving one or several data groups. Example: displaying an error message.
- **Basic FSM patterns:** A complete single COSMIC functional process.
- **Composite FSM pattern:** A set of basic FSM patterns having a high level functional meaning together. Example: The CRUDL (Create, Retrieve, Update, Delete, List) set of FPs to maintain data .
- **Multi-composite FSM pattern:** A set of composite and basic patterns having functional relationships among them.
  - In business application software, a multi-composite FSM pattern could represent a whole module, or component of a distributed application or even a whole application.
  - In embedded/real-time systems, it could be the set of back-end subsystem functionalities for a family of devices.

Functional Size Patterns

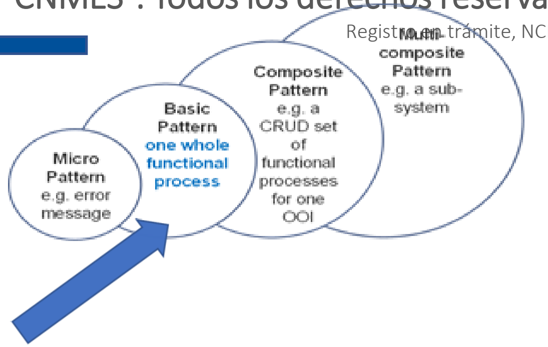

Multi-composite Pattern e.g. a sub-system
Composite Pattern e.g. a CRUD set of functional processes for one OOI
Basic Pattern one whole functional process
Micro Pattern e.g. error message

36 Example for **Micro FSM** pattern:

- Display simple error messages.

Functional Process	Data Group	Data Movements	Functional Size (in CFP)
<Functional process>	Error message	X	1
		Total:	1

Applicability & Reported Use

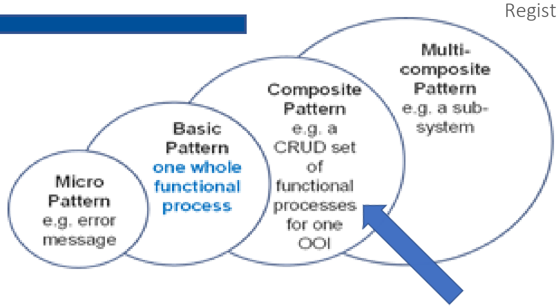


Example for a Basic Pattern:

- Create 1 data group

Functional Process	Data Group	Data Movements	Functional Size (in CFP)	Remark
Create <data group>	<data	ERW	3	Creates a new occurrence
	Error message	X	1	
Total:			4	

Functional Size Patterns

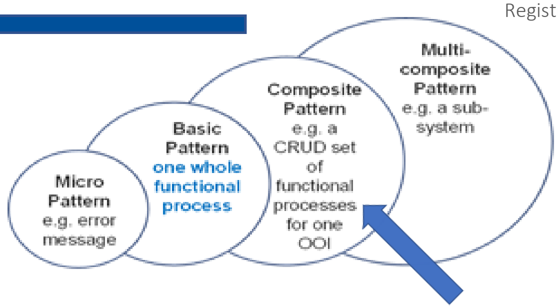


- CRUDL with 3 Data Groups

Functional Process	Data Group	Data Movements	Functional Size (in CFP)	Remark
Create <First DG>	<First DG>	ERW		Create new occurrence
	<Second DG>	RX		Read and display list
	<Third DG>	RX	2	Read and display list
	Error message	X	1	Subtotal: 8 CFP
Retrieve <First DG>	<First DG>	ERX	3	Select, read and display existing
	<Second DG>	RX	2	Must read its ID to display its name
	<Third DG>	RX	2	Same as above
	Error message	X	1	Subtotal: 8 CFP

Update <First DG>	<First DG>	ERW	3	Update existing occurrence
	<Second DG>	RX	2	Read and display list
	<Third DG>	RX	2	Read and display list
	Error message	X	1	Subtotal: 8 CFP
Delete a <First DG>	<First DG>	ERW	3	Delete an occurrence, reas it first, no other DG
	Message	X	1	Subtotal: 4 CFP
List <First DG>	<First DG>	RX	2	Read and display list
	Filter	E	1	Search filter applicable to all DGs
	<Second DG>	RX	2	Read/display list (filter)
	<Third DG>	RX	2	Same as above
	Error message	X	1	Subtotal: 8 CFP

Functional Size Patterns



Micro Pattern e.g. error message

Basic Pattern one whole functional process

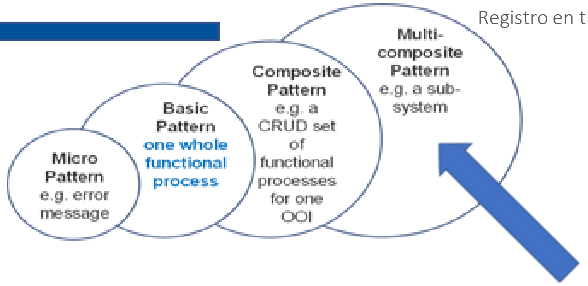
Composite Pattern e.g. a CRUD set of functional processes for one OOI

Multi-composite Pattern e.g. a sub-system

- CRUDL with 3 Data Groups

Functional Process	Data Group	Data Movements	Functional Size (in CFP)	Remark
Create <First DG>	<First DG>	ERW		Create new occurrence
	<Second DG>	RX		Read and display list
	<Third DG>	RX	2	Read and display list
	Error message	X	1	Subtotal: 8 CFP
Retrieve <First DG>	<First DG>	ERX	3	Select, read and display existing
	<Second DG>	RX	2	Must read its ID to display its name
	<Third DG>	RX	2	Same as above
	Error message	X	1	Subtotal: 8 CFP
		Total:	36	For this FSM pattern

Multi Composite Size Patterns


Micro Pattern e.g. error message
Basic Pattern one whole functional process
Composite Pattern e.g. a CRUD set of functional processes for one OOI
Multi-composite Pattern e.g. a sub-system

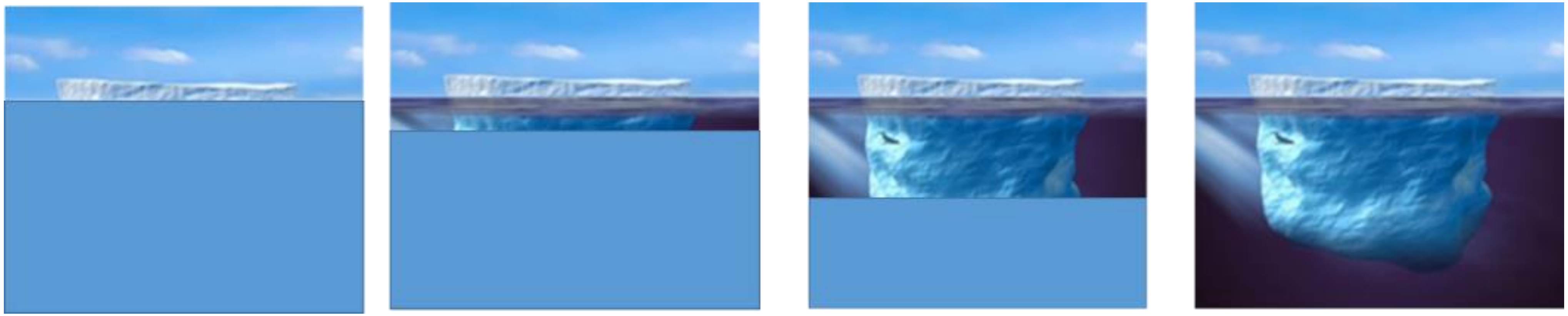
Example for a Module with 3 Data Groups

FSM Pattern	Category	Functional Size (in CFP)	Remark
CRUDL-3DG	Composite	36	Ex. for “Customer”
CRUDL-1DG	Composite	20	Ex. for “Sales Rep”
CRUDL-1DG	Composite	20	Ex. for “Customer category”
CRUD-2DG	Composite	22	Ex. for “Account aging parameters”
CRUD-3DG	Composite	26	Ex. for “Invoicing parameters”
CRUD-3DG	Composite	26	Ex. for “Cash receipt (C/R) parameters”
Transaction-7DG	Basic	12	Ex. for “Enter manual invoices”
Transaction-6DG	Basic	10	Ex. for “Enter a manual cash receipt”
Transaction-8DG	Basic	14	Ex. for “Enter adjustment on Invoice or C/R”
Report-3DG	Basic	7	Ex. for “Report on customer sales”
Report-4DG	Basic	9	Ex. for “Customer aging report”
Report-5DG	Basic	11	Ex. for “Customer statement of account”
Milestone-2DG	Basic	10	Ex. for “End of month A/R processing”
	Total:	223	For this FSM pattern

List of topics

1. Early Size Estimation issues
2. Size approximation techniques
3. The software-iceberg analoly for early sizing
4. Wrap-up

The iceberg analogy

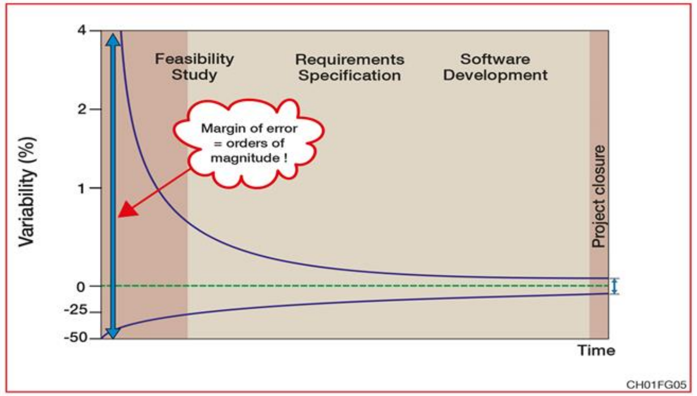


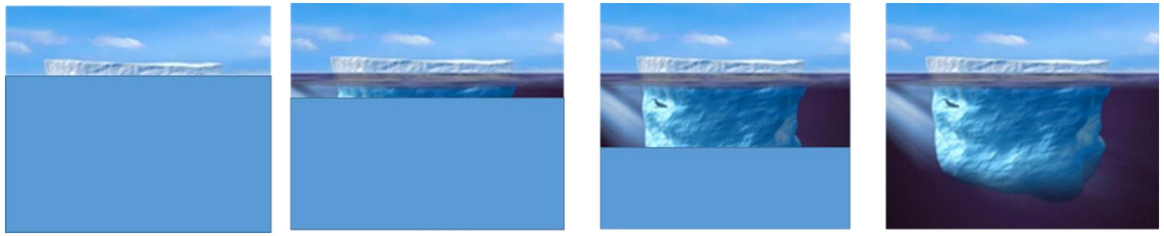
Initially visible requirements

Visibility increases

Additional sizing is required

Software-Iceberg Analogy (Abran & Vedadi)



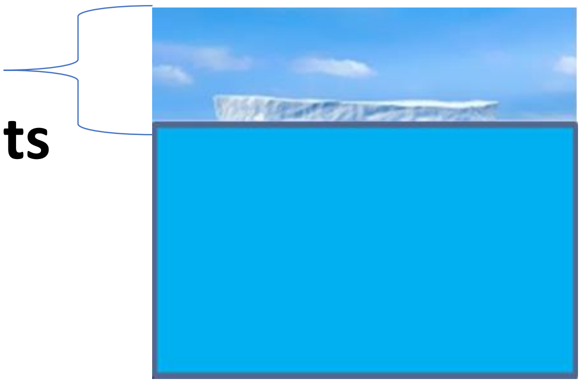


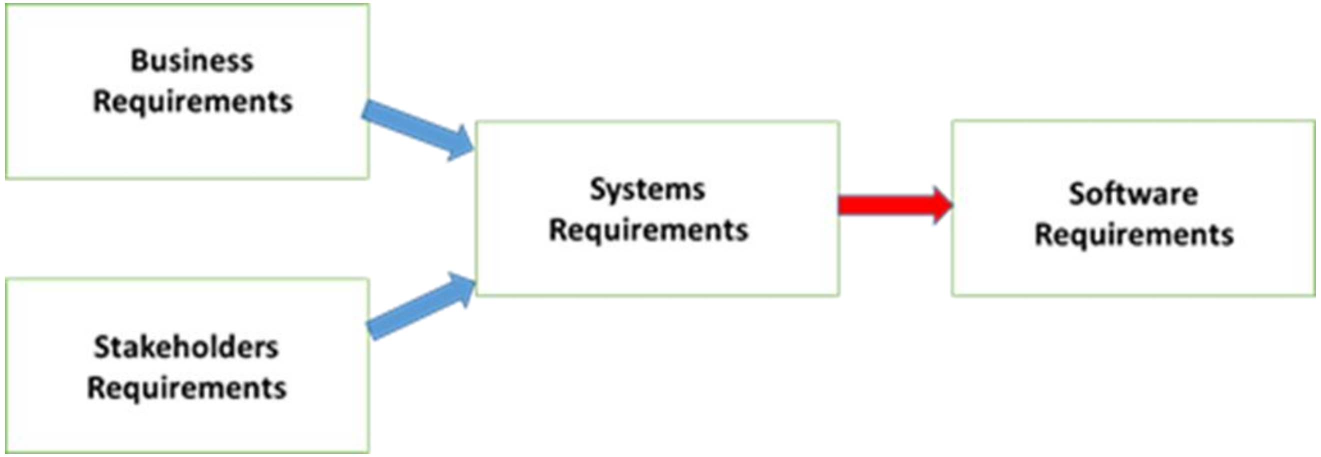
ISO-IEEE 25148

The ISO-IEEE 29148 on requirements engineering presents sources, types and levels of detail of the requirements throughout the system & software life cycle.

What is VISIBLE at Feasibility phase?

At feasibility & Early Requirements





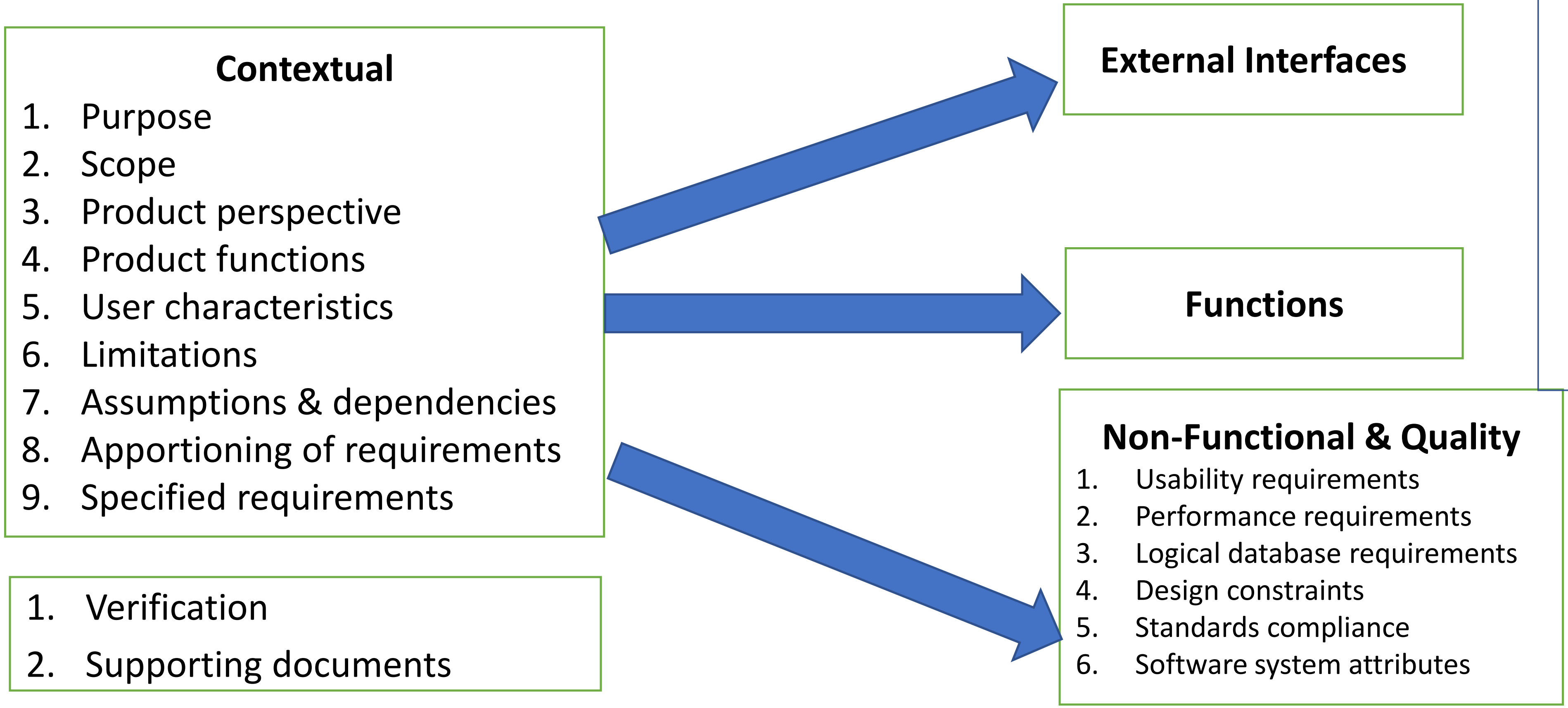
INTERNATIONAL STANDARD **ISO/IEC/ IEEE 29148**
First edition 2011-12-01

Systems and software engineering — Life cycle processes — Requirements engineering

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

ISO-IEEE 25148

What is visible at Early requirements Phase



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

Functions

Non-Functional & Quality

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

1. Verification
2. Supporting documents

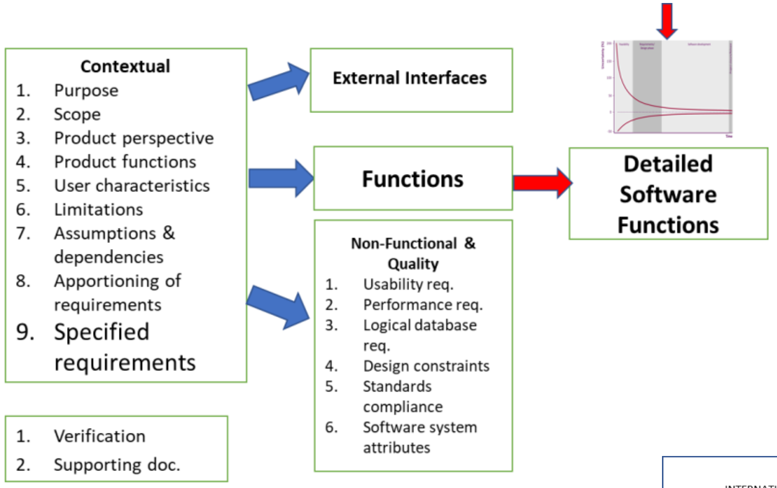
INTERNATIONAL STANDARD **ISO/IEC/ IEEE 29148**
First edition 2011-12-01

Systems and software engineering — Life cycle processes — Requirements engineering

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

ISO ISO-IEEE 25148

What is VISIBLE at End of Software Requirements Phase?



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.

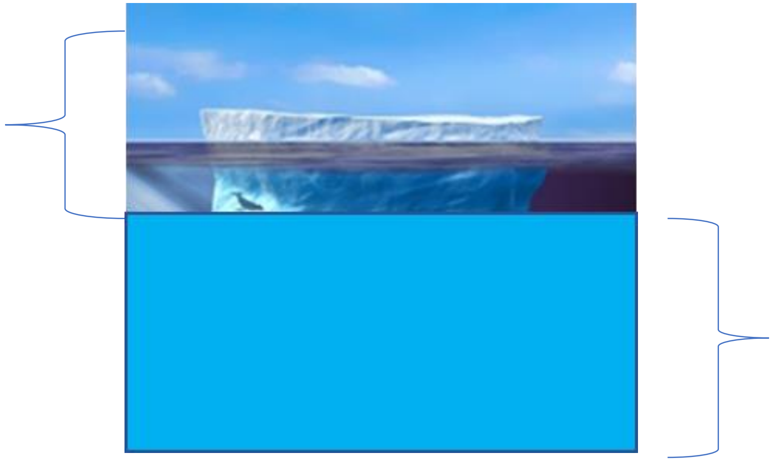
External Interfaces

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



The detailed software functions become visible

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

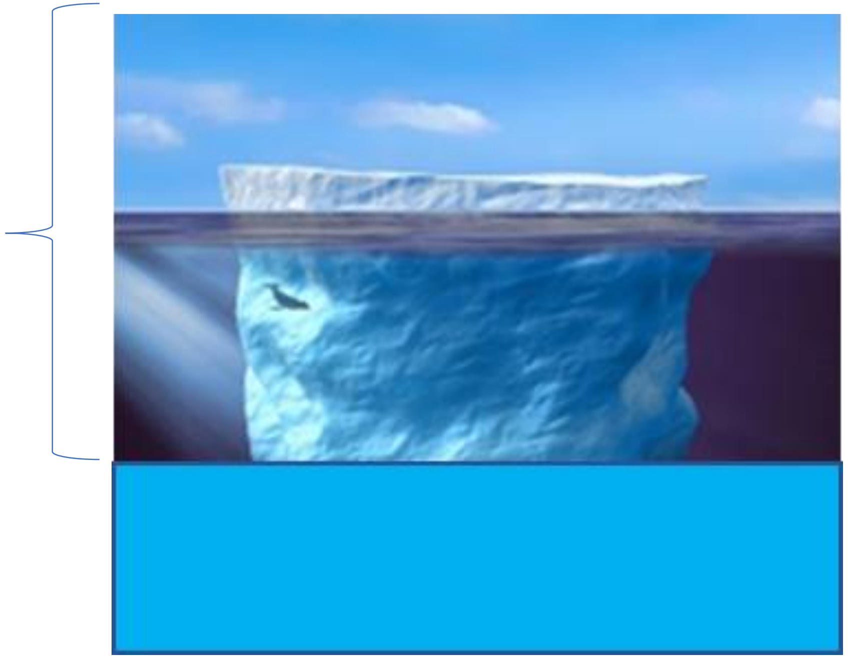
Systems and software engineering — Life cycle processes — Requirements engineering

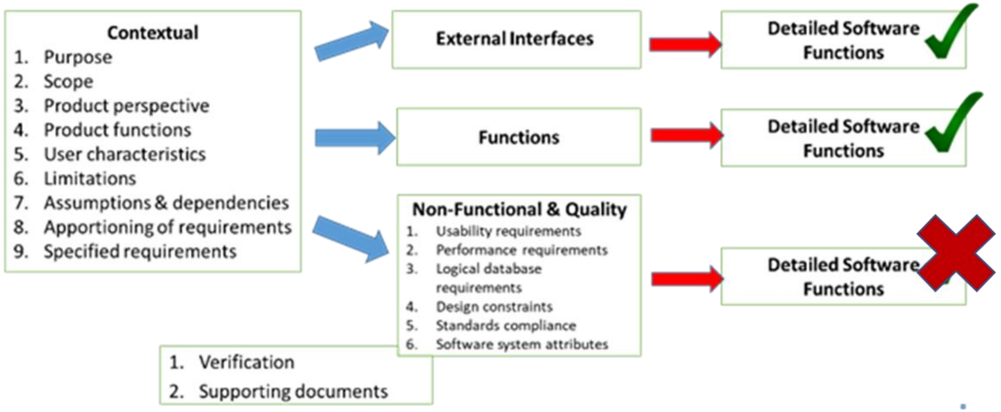
Ingénierie des systèmes et du logiciel — Processus du cycle de vie — Ingénierie des exigences

ISO-IEEE 25148

What Size is still NOT VISIBLE after requirements phase?

Functionality Specified & Approved





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 → Detailed Software Functions ✓

Functions → Detailed Software Functions ✓

Non-Functional & Quality
1. Usability requirements
2. Performance requirements
3. Logical database requirements
4. Design constraints
5. Standards compliance
6. Software system attributes

→ Detailed Software Functions ✗

What other software functions are not visible yet?

COSMIC

The COSMIC Functional Size Measurement Method
Version 4.0.1

Guideline on Non-Functional & Project Requirements

How to consider non-functional and project requirements in software project performance measurement, benchmarking and estimating

Version 1.
November 2015

Some of these NFR can be allocated to software & sized with COSMIC

Contextual

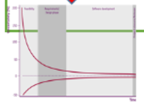
1. Purpose
2. Scope
3. Product perspective
4. Product functions
5. User characteristics
6. Limitations
7. Assumptions & dependencies
8. Apportioning of requirements
9. Specified requirements

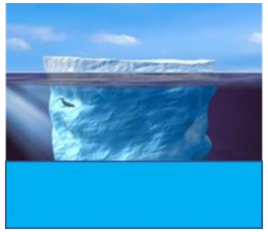
External Interfaces

Software Functions

Functions

Detailed Software Functions





COSMIC

The COSMIC Functional Size Measurement Method
Version 4.0.1

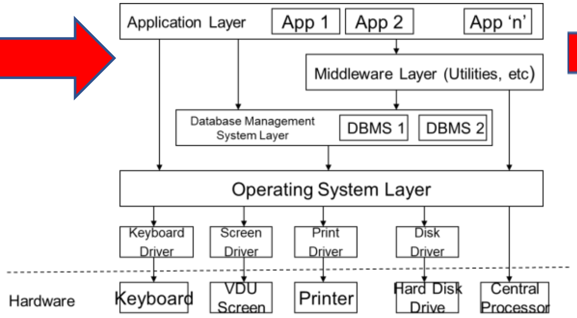
Guideline on
Non-Functional & Project
Requirements

How to consider non-functional and project requirements in software project performance measurement, benchmarking and estimating

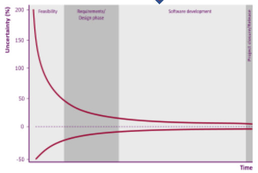
Version 1.
November 2015

Non-Functional & Quality

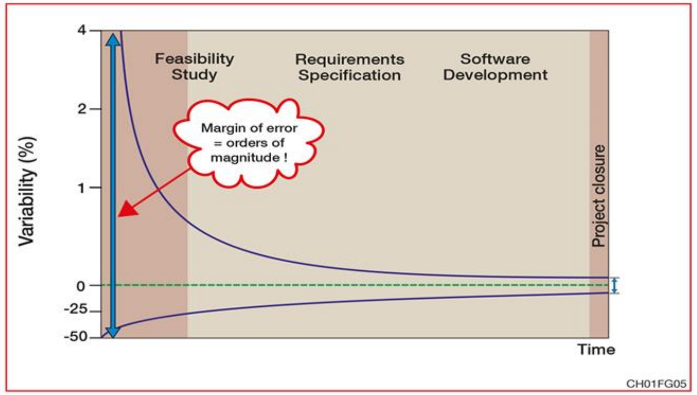
1. Usability req.
2. Performance req.
3. Logical database req.
4. Design constraints
5. Standards compliance
6. Software system attributes

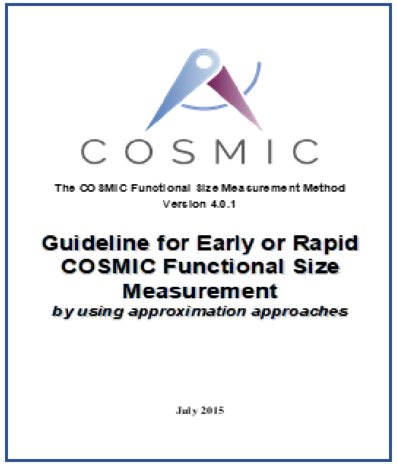


Software Functions



1. Verification
2. Supporting doc.





COSMIC

The COSMIC Functional Size Measurement Method
Version 4.0.1

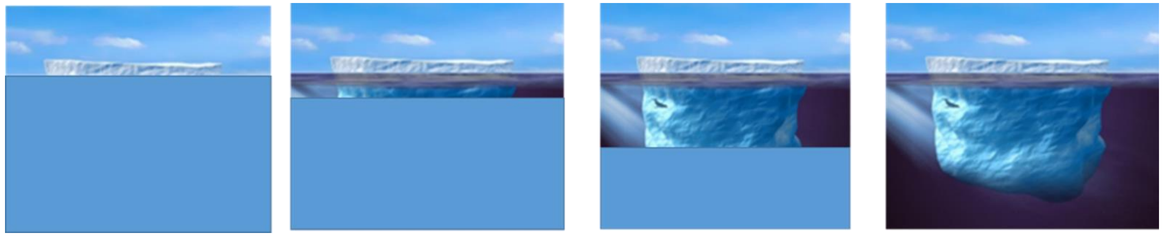
Guideline for Early or Rapid COSMIC Functional Size Measurement
by using approximation approaches

July 2015

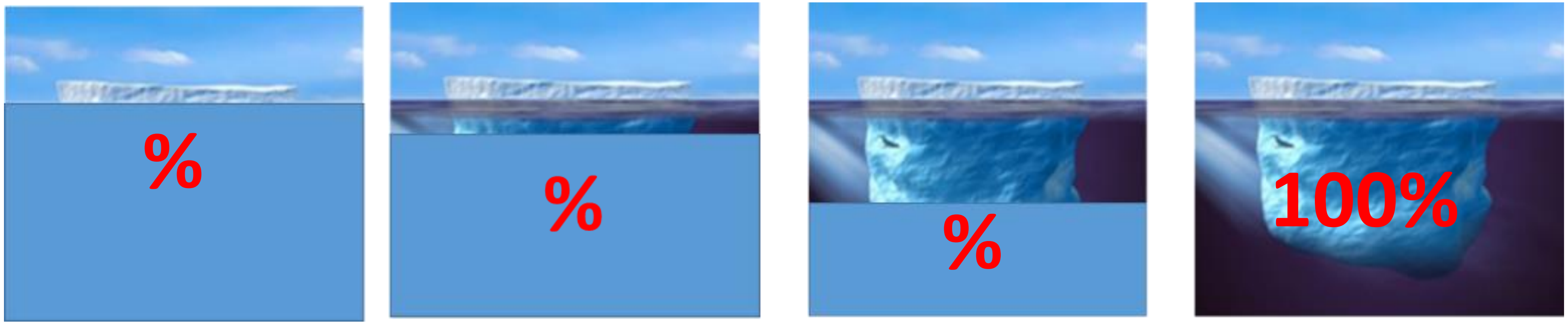
Upcoming 2019 Nov. Edition

ISO

ISO-IEEE 25148



The software iceberg analogy & sizing ratios



Level 1: Business functions → Level 2: Functions allocated to software → Level 3: Operational Functionality → Level 4: Quality & NFR allocated to software

COSMIC Case Studies

- Course Registration System (CRS) version 2.0.1
- Restaurant Management System version 1.1

COSMIC
www.cosmic-sizing.org

The concepts from ISO 29148 are used with these COSMIC case studies to:

- Identify types of requirements & when they become visible
- Identify COSMIC size ratios by types of requirements & phases
- Develop extrapolation ratios for future size estimation by phases and levels of documentation

Course Registration System - CRS

Level 1. Business functions (system functions)

- Information is available at Vision time or Feasibility phase

Level 2. Business functions allocated to software functional processes (software functions)

- Often available early in the specification phase

Level 3. Detailed functionality allocated to each software functional process

- At the coding phase, up to final testing

Level 1: business functions - CRS

NO	Business function
1	Maintain professor information (by the registrar)
2	Maintain student information (by the registrar)
3	Maintain courses to teach (by professor)
4	Maintain student schedule (by students)
5	Close registration (by the registrar)
6	Submit grades (by professor)
7	Enquire report card (by students)

Level 2: Software functions – CRS

NO	Functional process
1	Add a professor
2	Modify a professor
3	Delete professor
4	Enquire on a professor
5	Add a student
6	Modify a student
7	Delete a student
8	Enquire on student
9	Add courses to teach
10	Modify a course to teach
11	Delete courses to teach
12	Enquire courses to teach
13	Enquire on course to teach details
14	Add courses
15	Modify a course
16	Delete a course
17	Enquire on courses
18	Enquire on course details
19	Close registration
20	Submit grades
21	Enquire on a report card

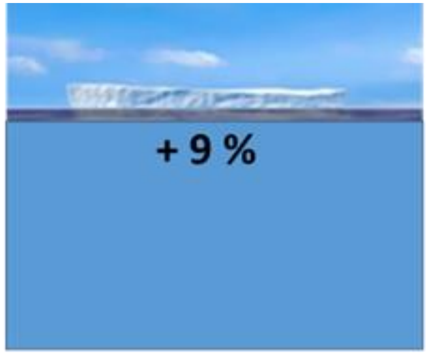
Classification results – CRS case study

CRS Cases study: Total functional size: 102 CFP

Direct business function	**Business details**	**Operational business**	**Quality NFR**	**Interface**
21 CFP	9 CFP	42 CFP	30 CFP	1 CFP
Percentage over total size				
20 %	**9 %**	**41 %**	**30 %**	**-**

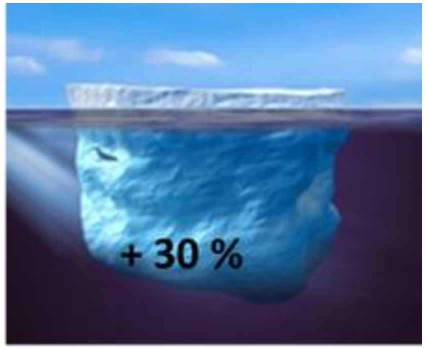
CRS Case study: Classification results

Systems Functions



Details of Business Functions

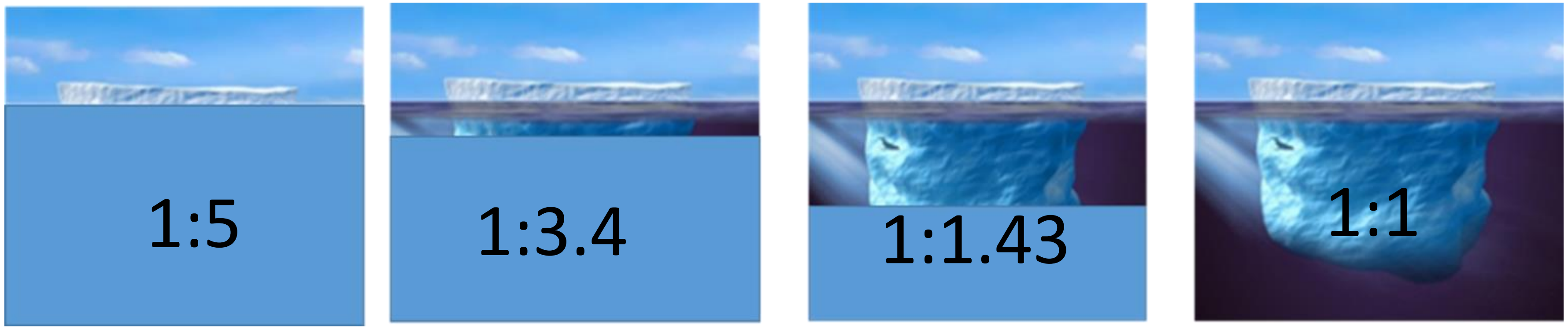
Operational Functions



Quality Functions

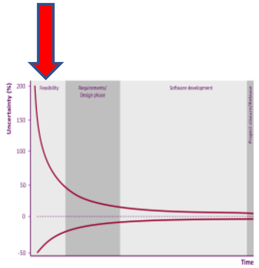
- 21 % system functions
- 9 % detail functions
- 42 % operational functionality
- 30 % implementation of quality (data integrity)

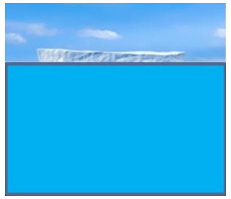
Transformation into Scaling factors of Requirements

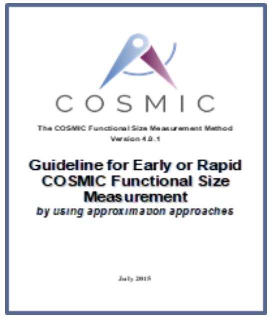


- ➢ System functions: 20% leads to a 1:5 scaling factor
  - ✓ Example: a size of 10 FP would lead to 10x5= 50 CFP when fully specified, including operational functions and data integrity functions.
- ➢ Detail functions: 20%+9% (= 29%) leads to a 1:3.4 scaling factor
  - ✓ a size of 20 CFP would lead to 20CFP x 3.4 = 68 CFP
- ➢ Operational functionality: 20%+9%+41%= 70% leads to a 1:1.43 scaling factor
  - ✓ a size of 20 CFP would lead to 20CFP x 1.43 = 29 CFP

Guidelines for Size Estimation through the lifecycle







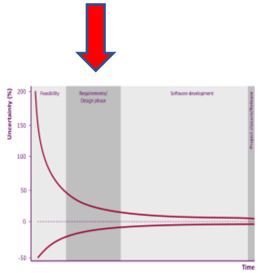
COSMIC

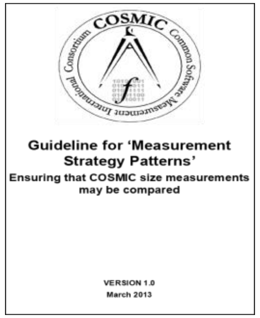
The COSMIC Functional Size Measurement Method
Version 4.0.1

Guideline for Early or Rapid COSMIC Functional Size Measurement

by using approximation approaches

July 2015





Guideline for 'Measurement Strategy Patterns'

Ensuring that COSMIC size measurements may be compared

VERSION 1.0
March 2013

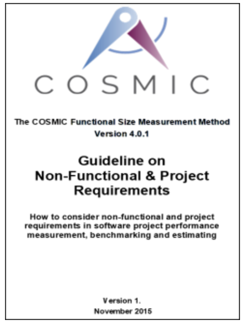
INTERNATIONAL STANDARD ISO/IEC/ IEEE 29148
First edition 2011-12-01

Systems and software engineering — Life cycle processes — Requirements engineering

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

ISO

ISO-IEEE 25148



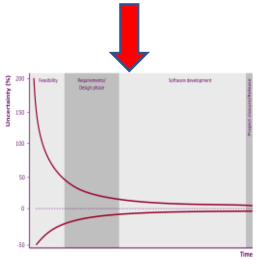
COSMIC

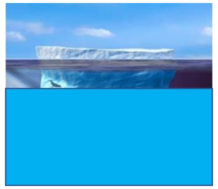
The COSMIC Functional Size Measurement Method
Version 4.0.1

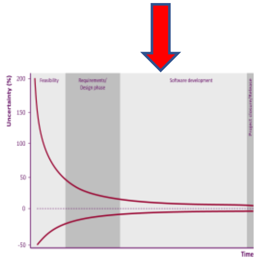
Guideline on Non-Functional & Project Requirements

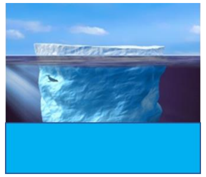
How to consider non-functional and project requirements in software project performance measurement, benchmarking and estimating

Version 1.
November 2015









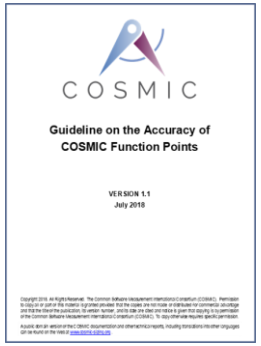
COSMIC

The COSMIC Functional Size Measurement Method
Version 4.0.1

Measurement Manual

(The COSMIC Implementation Guide for ISO/IEC 19761: 2011)

April 2015



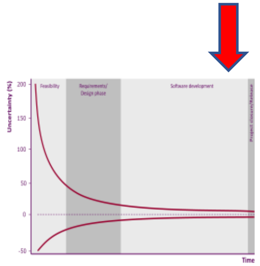
COSMIC

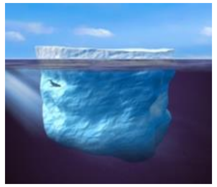
Guideline on the Accuracy of COSMIC Function Points

VERSION 1.1
July 2018

Copyright 2018. All Rights Reserved. The Common Software Measurement International Consortium (COSMIC). Permission to copy all or part of this material is granted provided that the copies are not made or distributed for commercial advantage and that the title of the publication, its version number, and its date are cited and notice is given that copying is by permission of the Common Software Measurement International Consortium (COSMIC). To copy otherwise requires specific permission.

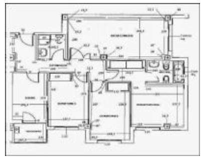
A public domain version of the COSMIC documentation and other technical reports, including translations into other languages can be found on the Web at www.cosmic-sizing.org.





Future Impact of software-Iceberg analogy technique

1. Set of scaling factors should be added in the description of requirements as a ‘best practice’ in requirements engineering.



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

Systems and software engineering — Life cycle processes — Requirements engineering

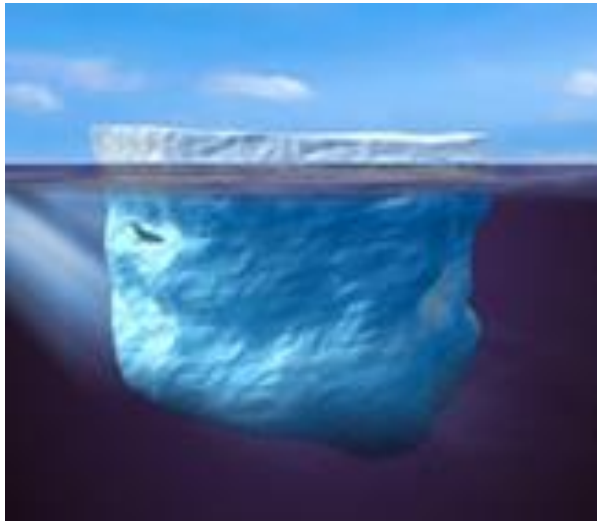
Ingénierie des systèmes et du logiciel — Processus du cycle de vie — Ingénierie des exigences

ISO ISO-IEEE 25148

2. Scaling ratios, with successive levels of documentation, can be used in future projects as project progress through the lifecycle.

3. These scaling factors can provide also information on the levels of completeness of the documentation of the requirements.

From simple to complex contexts in software



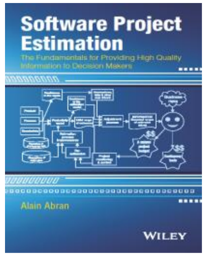
List of topics

1. Early Size Estimation issues
2. Size approximation techniques
3. The software-iceberg analoly for early sizing
4. Wrap-up

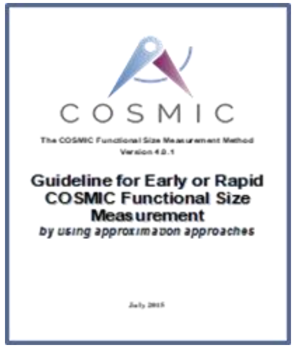
Upcoming related certification initiatives

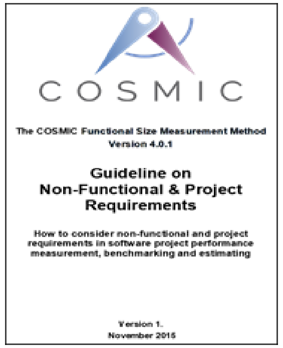
1. ICEAA: Software Cost Estimation – A broad view

ICEAA International Cost Estimating & Analysis Association


Software Project Estimation
The Fundamentals for Providing High Quality Information to Decision Makers
Alain Abran
WILEY

2. COSMIC – Size Estimation – A specialized view


COSMIC
The COSMIC Functional Size Measurement Method
Version 4.0.1
Guideline for Early or Rapid COSMIC Functional Size Measurement
by using approximation approaches
July 2015


COSMIC
The COSMIC Functional Size Measurement Method
Version 4.0.1
Guideline on Non-Functional & Project Requirements
How to consider non-functional and project requirements in software project performance measurement, benchmarking and estimating
Version 1.
November 2015

ECORFAN®