COSMIC

SOFTWARE PROJECTS ESTIMATION & CONTROL: VERSATILITY & CONTRIBUTIONS OF COSMIC FUNCTION POINTS

Alain Abran
with C. Symons, C.Ebert, F.Vogelezang, H.Soubra

ICEAA 2017 Professional Development & Training Workshop
Portland, Oregon (USA), June 6-9, 2017

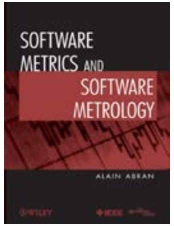
Presenter background - Alain Abran

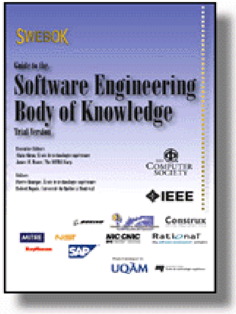
20 years

- Development
- Maintenance
- Process Improvement

20 years

+ 40 PhD





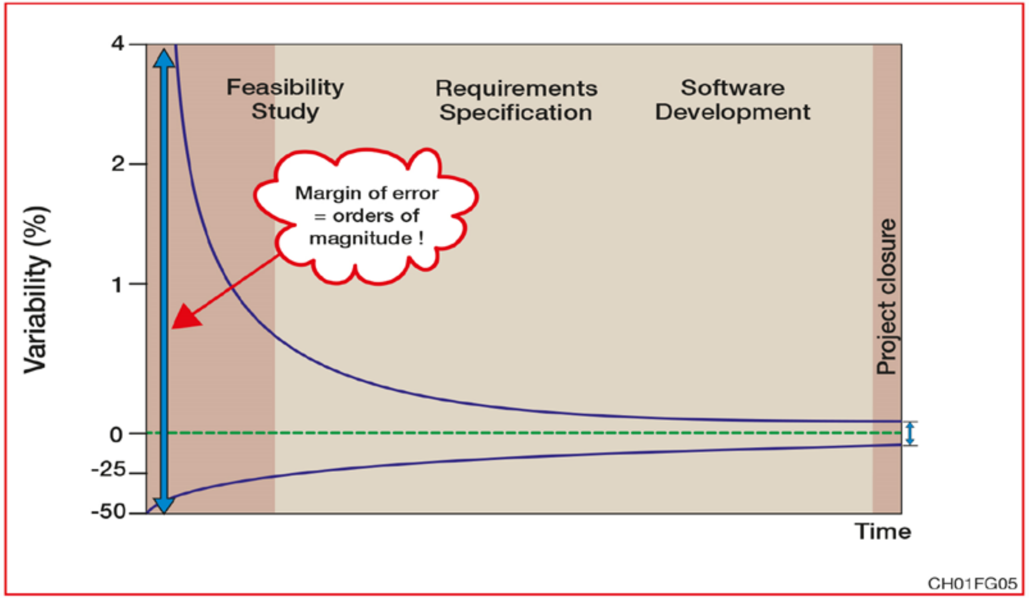
ISO: 19761, 9126, 25000, 15939, 14143, 19759

COSMIC

Agenda

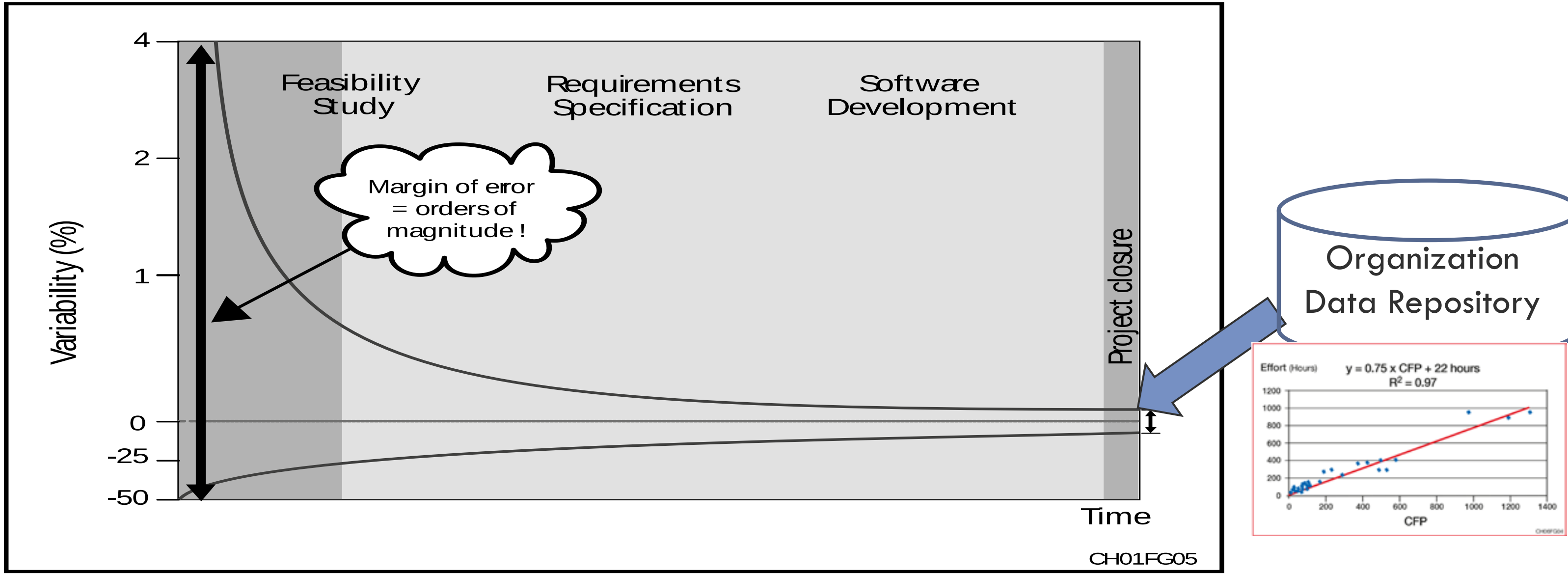
1. Software effort estimation & software size
2. COSMIC: 2nd generation of Function Points
3. Versatility of COSMIC Function Points
4. Contributions of COSMIC to Estimation models
5. Early & Quick COSMIC sizing at estimation time
6. Automation of COSMIC Function Points
7. Summary

The Cone of Uncertainty across the Project Lifecycle



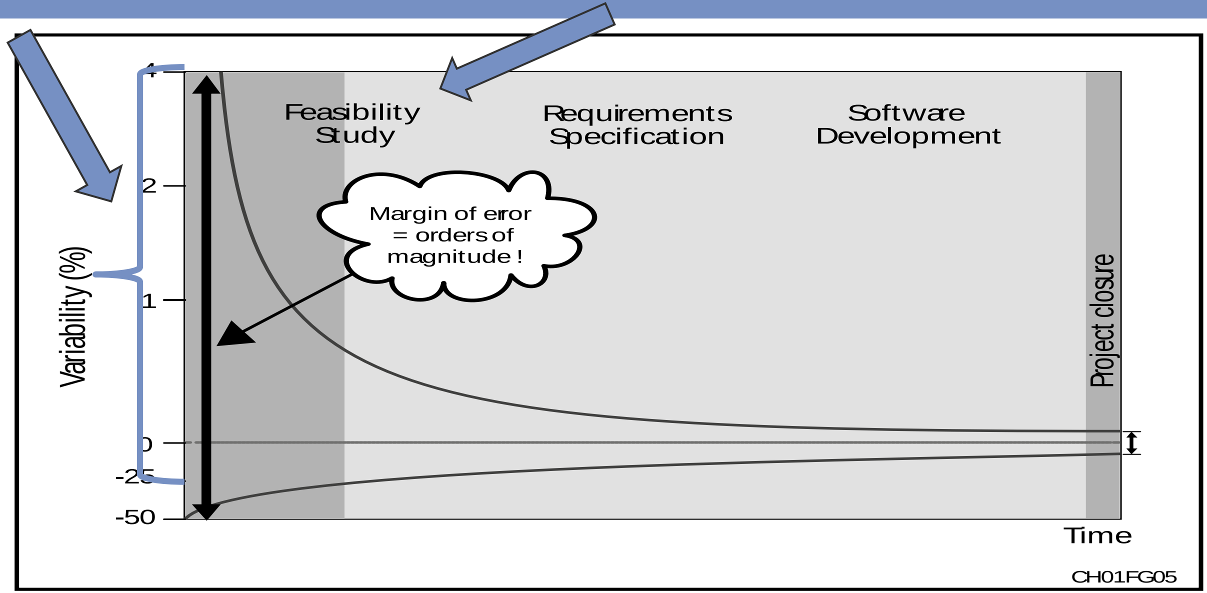
Range of expected variations in 'estimation' models across the project life cycle
Adapted from Boehm (2000), Fig. 1.2

Productivity Models: Built with Data from Completed Objects

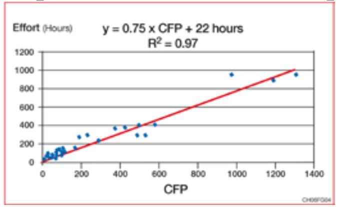


You build estimation models with completed projects (with almost no uncertainty in the inputs)

Estimation Foundations: Productivity Models with Uncertainties in the Inputs

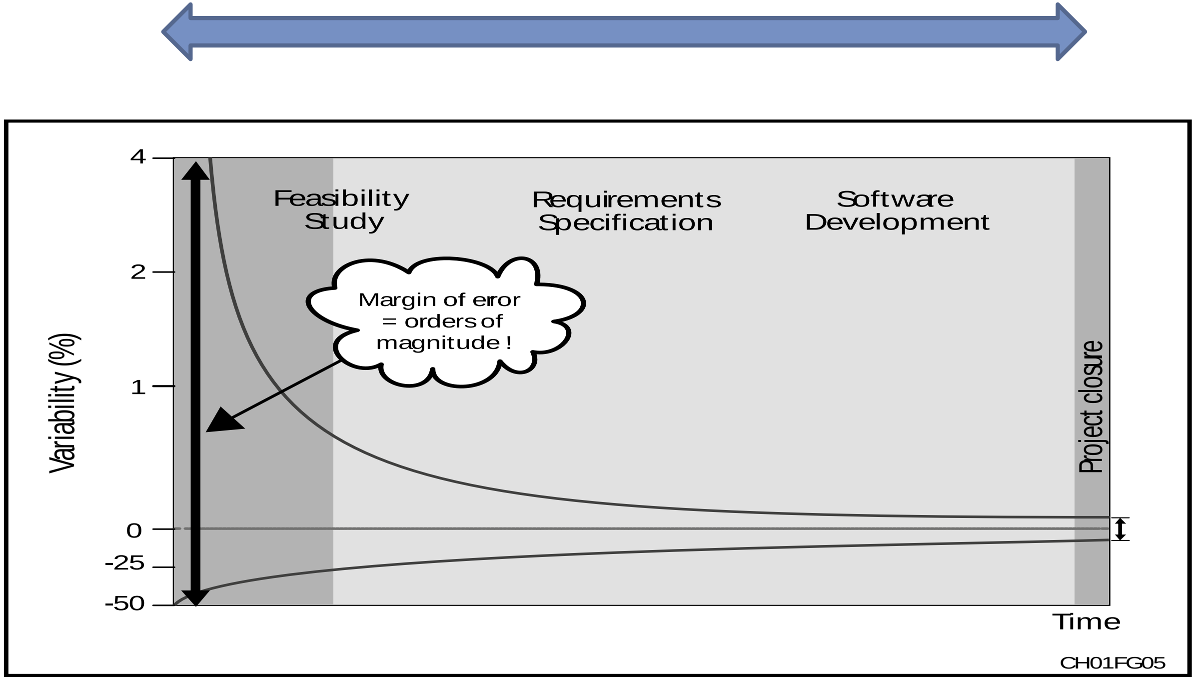


Organization Data Repository



- **You do estimation upfront with a lot of uncertainty**
- **What do you have upfront as available for estimation purposes?**

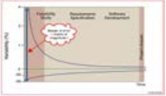
Software: What is Available & Measurable Across the Lifecycle?



Software Sizing Options across the Lifecycle?

Sizing method options:

- Lines of code:
  - X Can't estimate until software designed
  - X Technology-dependent, no standards



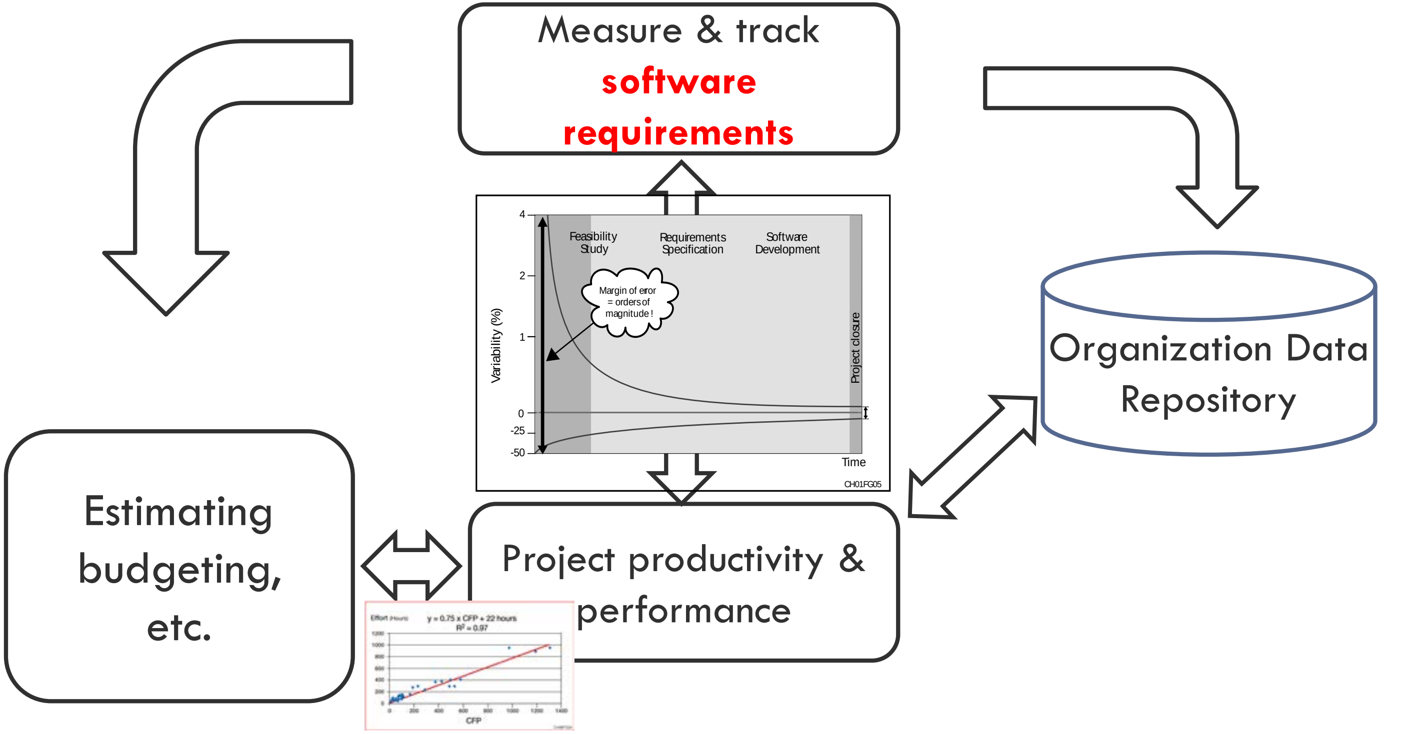
- Usecase Points, Object Points, ..
  - X Technology dependent, no standards,
  - X Mathematical validity?

- Story Points
  - X Entirely subjective

- Functional size
  - ✓ (Function Points):
  - ✓ International standard methods
  - ✓ Technology-independent

ISO International Organization for Standardization

Measurable Software Requirements: the Foundations for Estimation & Benchmarking Studies



Agenda

1. Software Effort Estimation & Software Size
2. COSMIC: 2nd generation of Function Points
3. Versatility of COSMIC Function Points
4. Contributions of COSMIC to Estimation models
5. Early & Quick COSMIC sizing at estimation time
6. Automation of COSMIC Function Points
7. Summary

1^{st} Generation of Function Points: Weights

FTR's	DATA ELEMENTS		
	1-4	5-15	> 15
0-1	Low	Low	Ave
2	Low	Ave	High
3 or more	Ave	High	High

Inputs - Matrix

FTR's	DATA ELEMENTS		
	1-5	6-19	> 19
0-1	Low	Low	Ave
2-3	Low	Ave	High
> 3	Ave	High	High

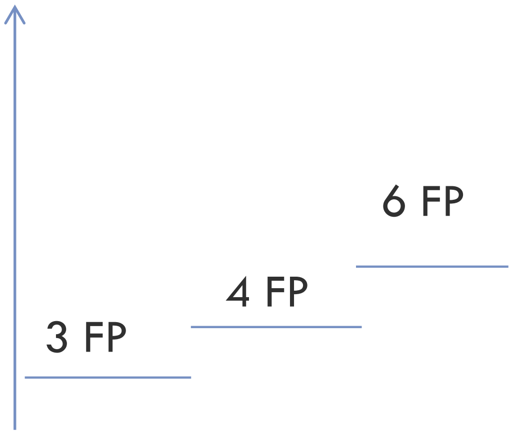
Output & Enquiries – Shared Matrix

Rating	VALUES		
	EO	EQ	EI
Low	4	3	3
Average	5	4	4
High	7	6	6

Transactions: weights in FP (Function Points)

1st Generation of Function Points: Step Functions

Function Points (FP)

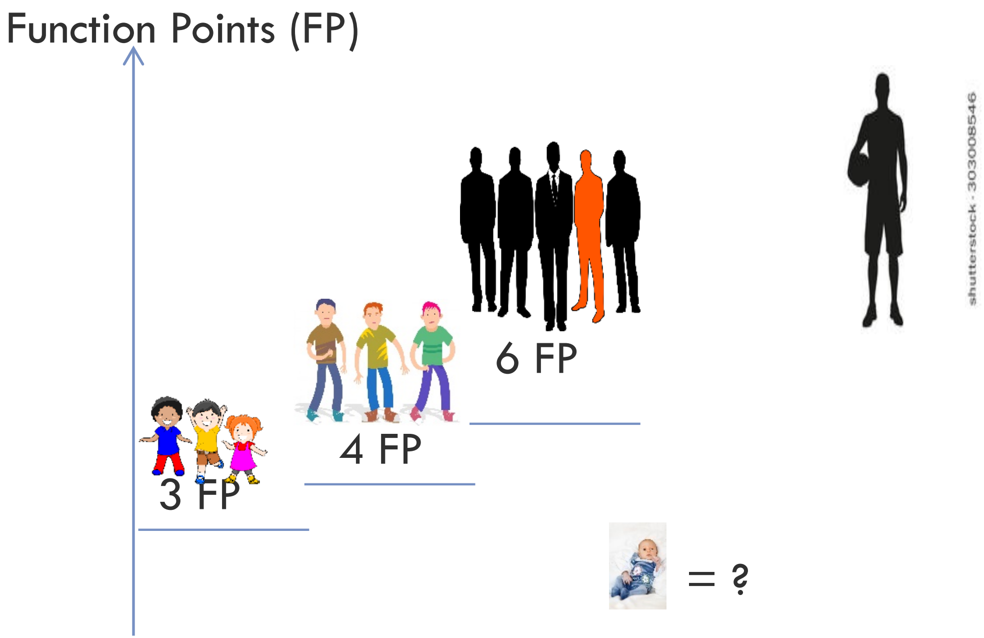


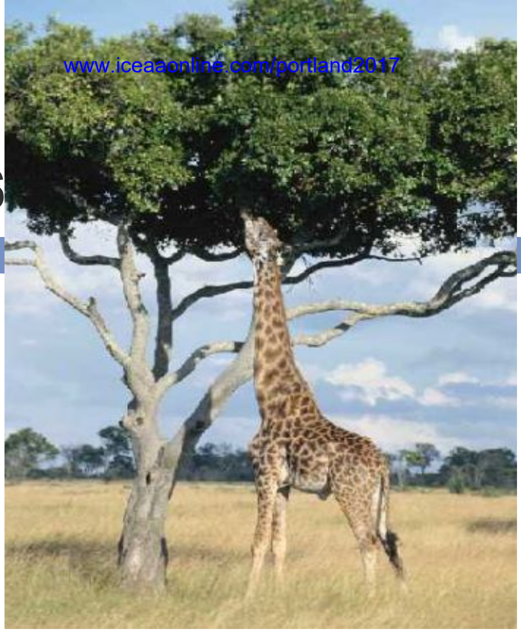
Key limitations:

- **Only 3 values**
- **Limited ranges (min,max)**
- **No single measurement unit of 1 FP!**

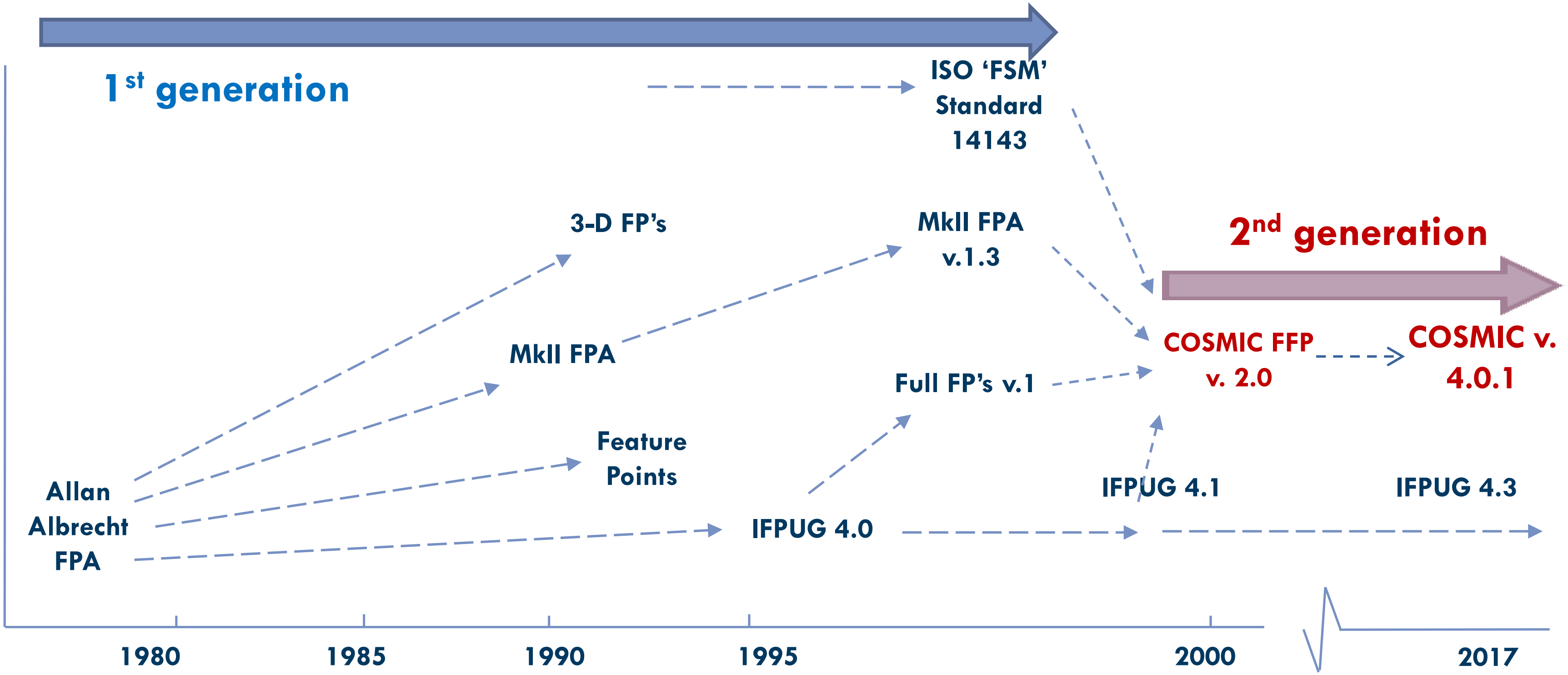
3-step size range for the IFPUG External Input Transactions

1st Generation of Function Points





1st & 2nd generation of Function Points



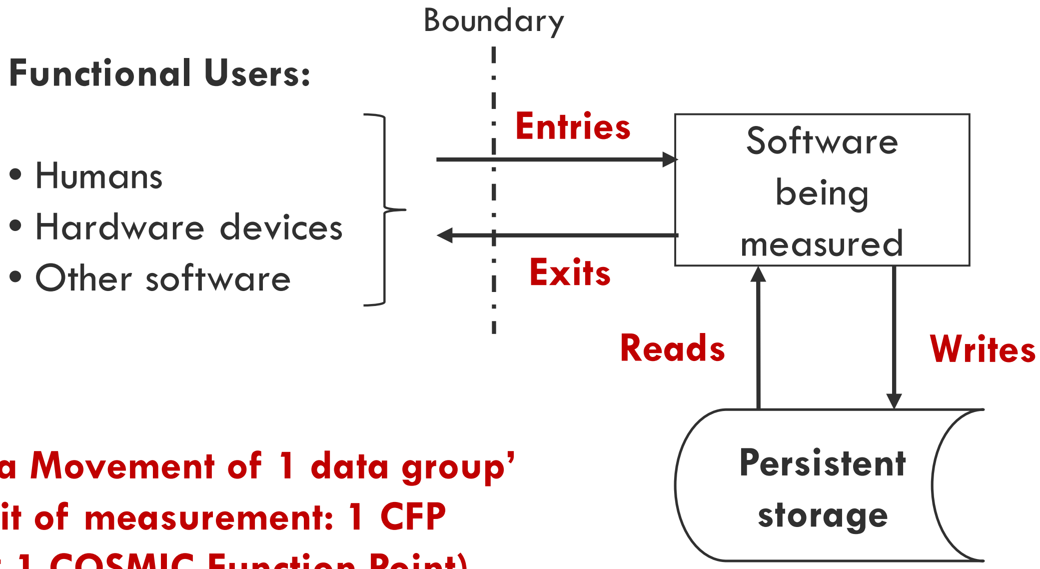
COSMIC

2nd Generation of Function Points

Every software is different, but what is common:

- In all software?
  - In different types of software?
  - In very small or extremely large software?

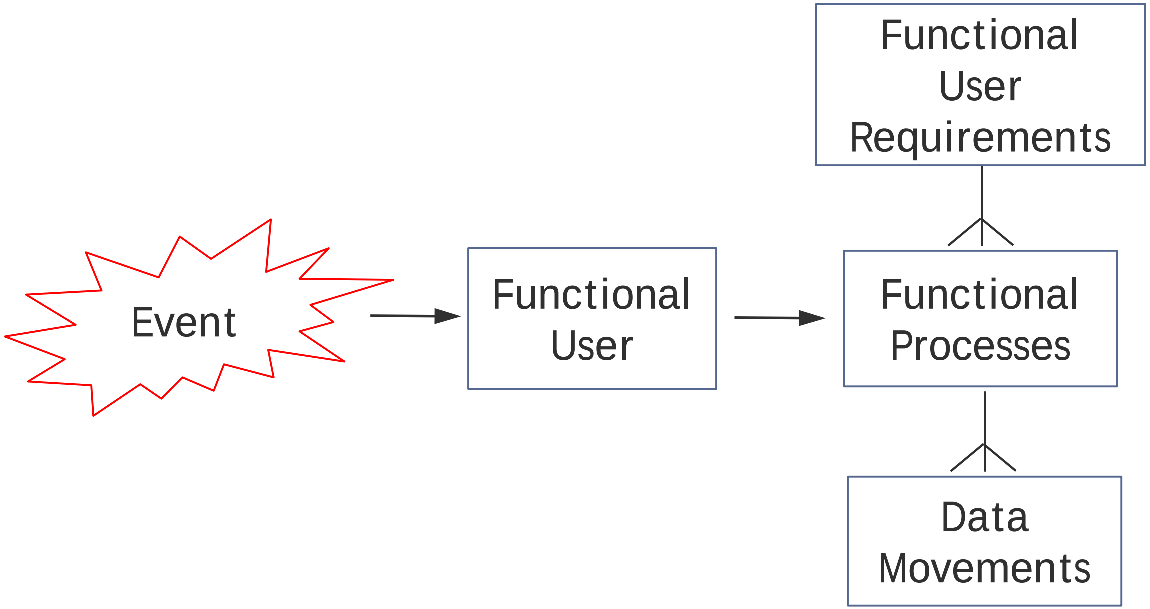
2nd Generation of Function Points: All software does this!



COSMIC view of software

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

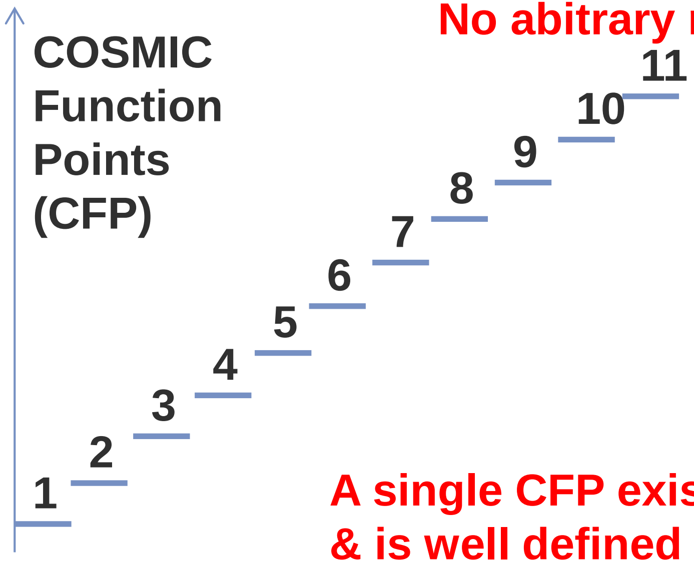
All Software also does this…



COSMIC view of software

2nd Generation with COSMIC

No abitrary max



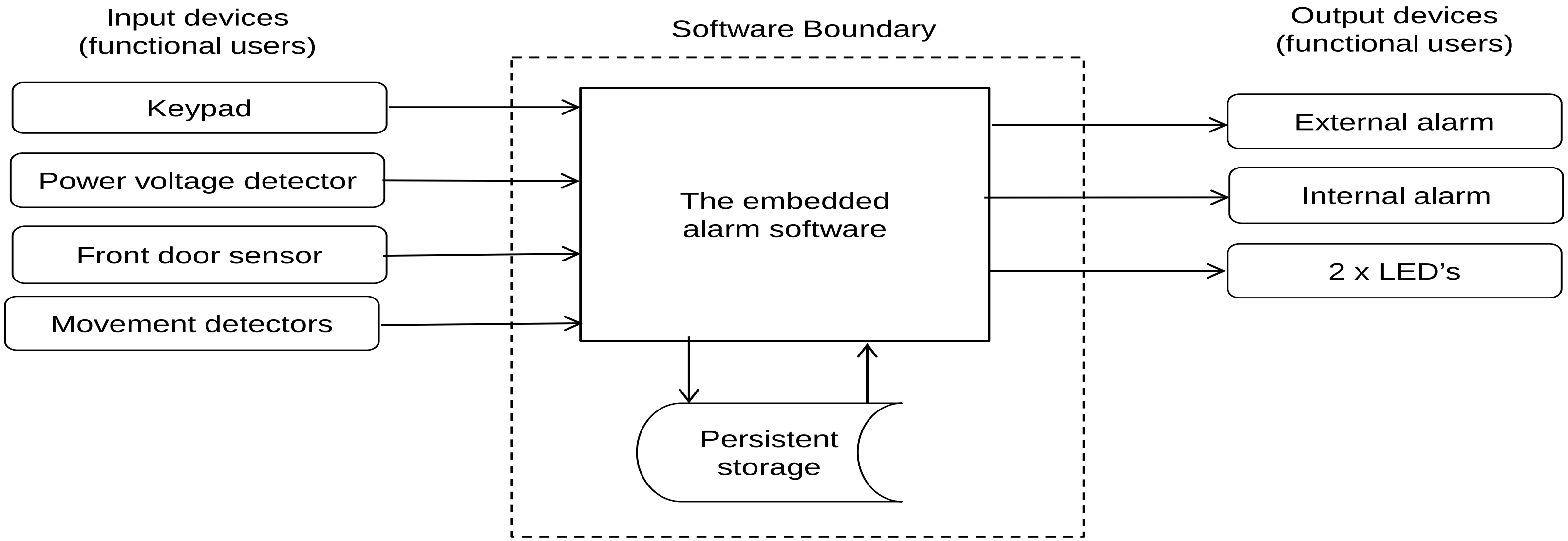
A single CFP exists & is well defined

Largest observed functional processes:

In avionics >100 CFP

In banking > 70 CFP

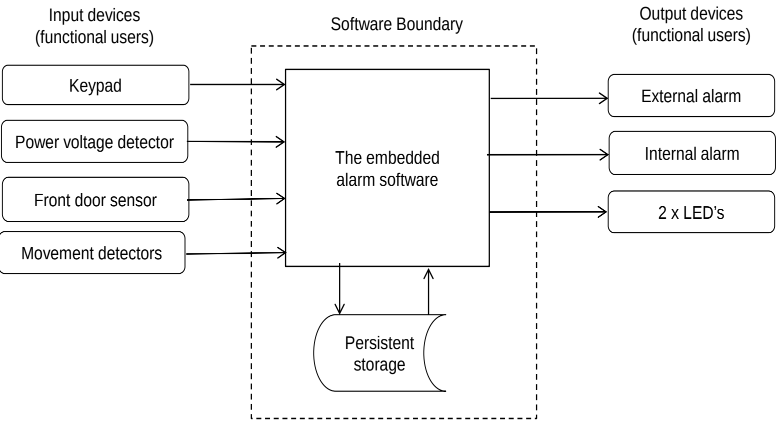
Example 1: Intruder Alarm System - Requirements



Example 1: Intruder Alarm System – COSMIC size

Functional process: Possible intruder detected.

Triggering event: Door opens whilst alarm system is activated.

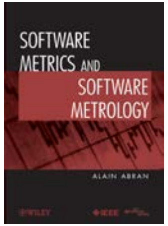


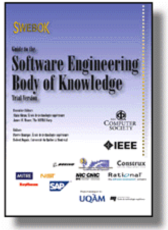
Data Movement	Functional User	Data Group
Entry	Front-door sensor	'Door open' message (triggering Entry)
Read	- / Occupant	PIN (from persistent storage)
Exit	Green LED	Switch 'off' command
Exit	Red LED	Switch 'on' command
Exit	Internal siren	Start noise command
Entry	Keypad	PIN (If the wrong code is entered, the user may enter the PIN two more times but the process is always the same so it is only measured once.)
*	Green LED	Switch 'on' command (after successful entry of PIN)
*	Red LED	Switch 'off' command
Exit	Internal siren	Stop noise command (after successful entry of PIN)
Exit	External siren	Start noise command (after three unsuccessful PIN entries, or if the PIN is not entered in time)
Exit	External siren	Stop noise command (after 20 minutes, a legal requirement)

Size = 9 CFP (COSMIC Function Points)

In summary: COSMIC Function Points

- Designed by an international group of software measurement experts
  - COSMIC: Common Software Measurement International Consortium
- To measure the Functional User Requirements of:
  - Business applications
  - Real-time
  - Infrastructure software
  - Various other types of software
  - Hybrids of these
- Based on:
  - Metrology
  - Fundamental software engineering principles
- An ISO standard: ISO 19761
- Open, freely available (via www.cosmic-sizing.org)





Agenda

1. Software Effort Estimation & Software Size
2. COSMIC: 2nd generation of Function Points
3. Versatility of COSMIC Function Points
4. Contributions of COSMIC to Estimation Models
5. Early & Quick COSMIC sizing at Estimation Time
6. Automation of COSMIC Function Points
7. Summary

Versatility - Guidelines by Application Domains

- Business applications
- Real-time software
- Data Warehouse software
- SOA software (SOA: Service Oriented Architecture)
- Mobile apps
- Agile Development


COSMIC

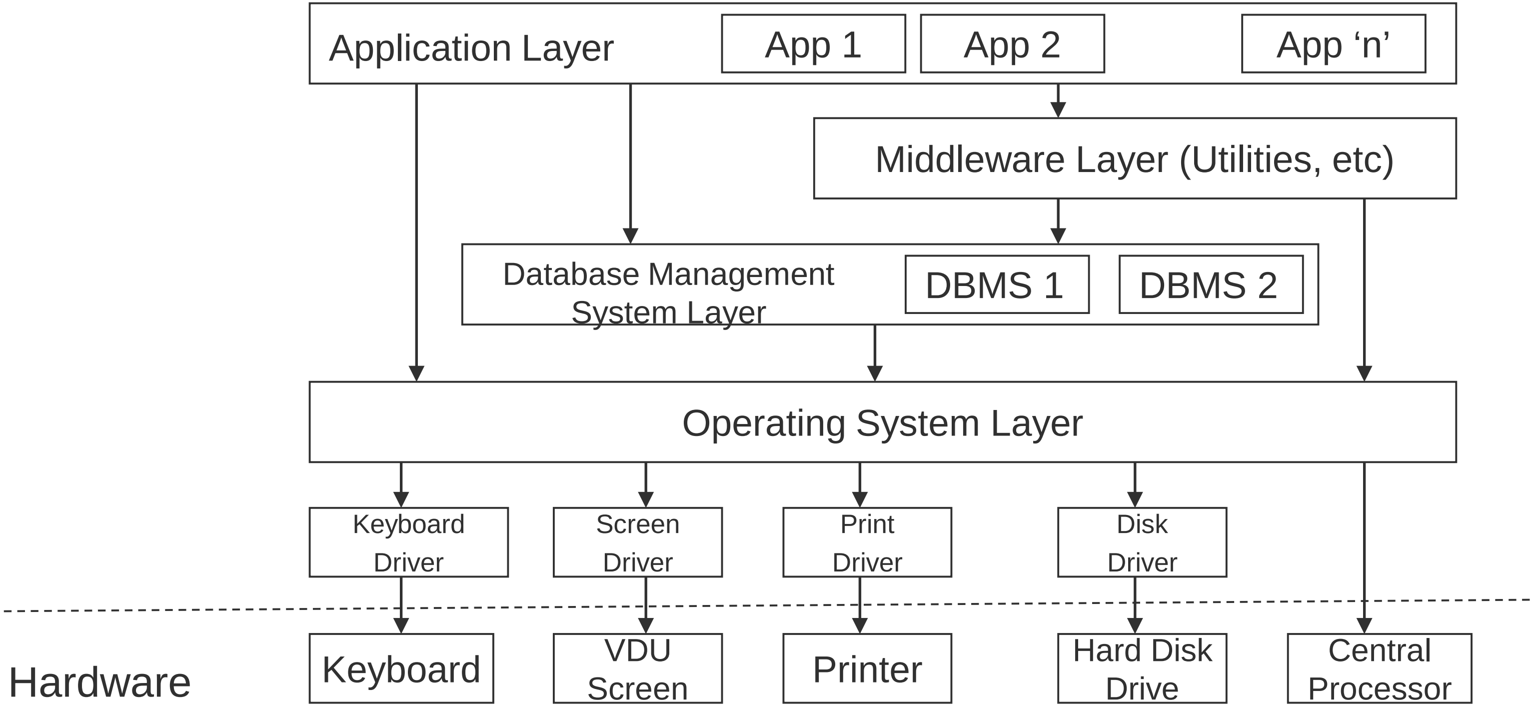
The COSMIC Functional Size Measurement Method
Version 4.0.1

Guideline for Sizing
Business Application Software

VERSION 1.3a
Febuary 2016

COSMIC

Versatility - at any level of software requirements



Versatility – COSMIC Case Studies

- Real-time:
  - Rice cooker
  - Automatic line switching
  - Valve control
- Business:
  - Course registration (distributed)
  - Restaurant management (web & mobile phone)
  - Banking web advice module
  - Car hire (existing legacy app.)

COSMIC

Agile: Aggregation rules for components, sprints, etc. up to whole software systems

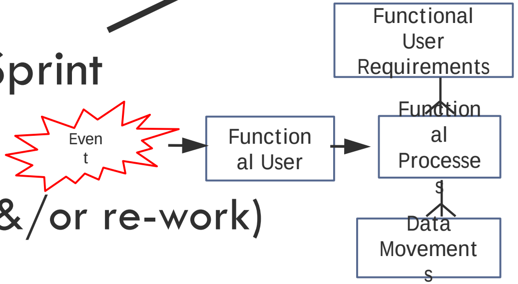
System

Release

Iteration

Sprint

User Story (new &/or re-work)



COSMIC size usable for:

- early Total System sizing & effort estimation
- US, Sprint etc. sizing & estimation
- Progress control at any level

COSMIC

Agenda

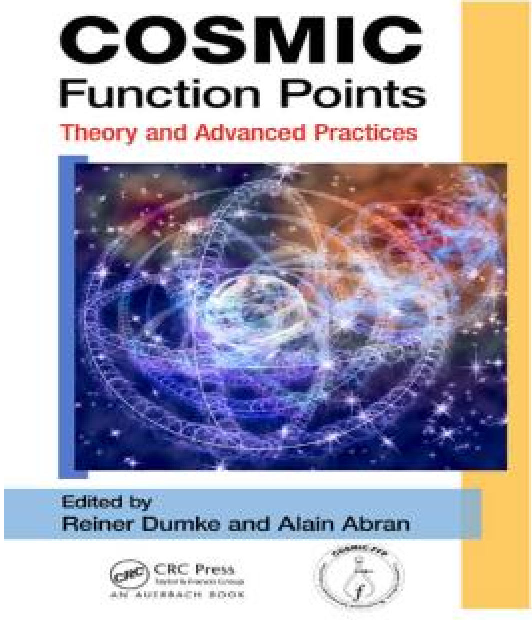
1. Software Effort Estimation & Software Size
2. COSMIC: 2nd generation of Function Points
3. Versatility of COSMIC Function Points
4. Contributions of COSMIC to Estimation models
5. Early & Quick COSMIC sizing at estimation time
6. Automation of COSMIC Function Points
7. Summary

COSMIC data from Industry

Practical experimentations with the COSMIC method in Automotive embedded software field

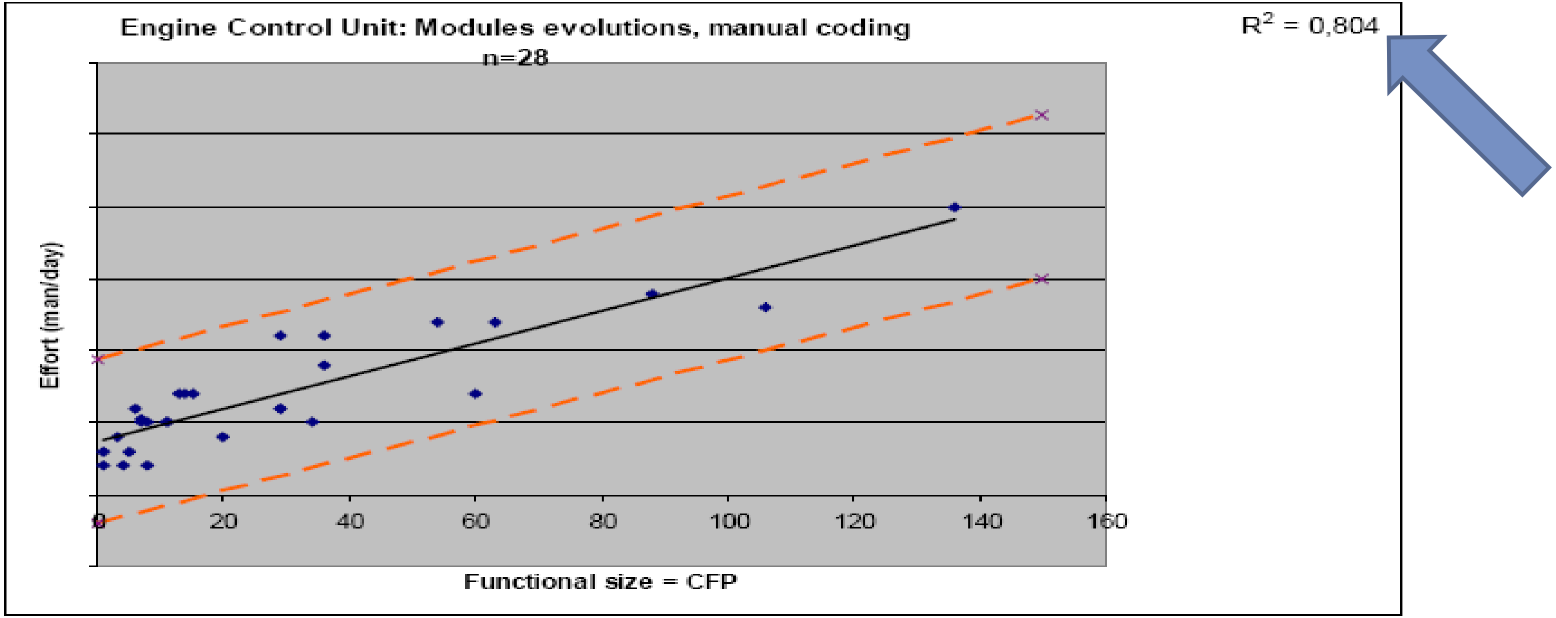
By: Sophie Stern

Renault



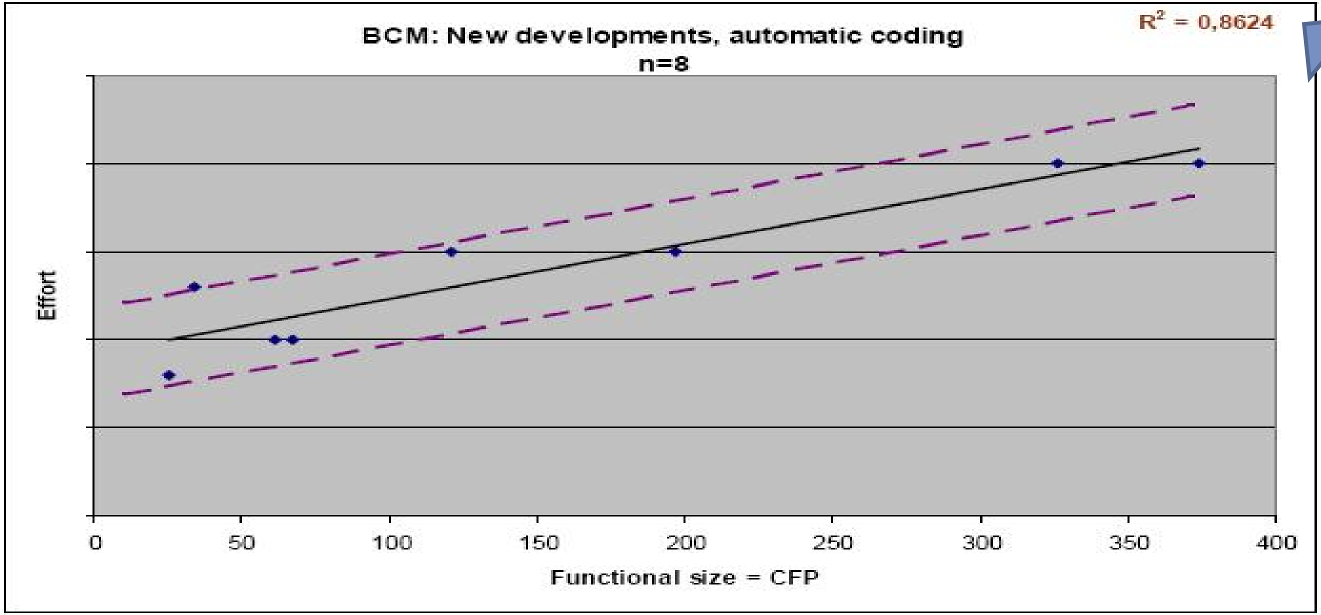
COSMIC

Data from Renault - 2012

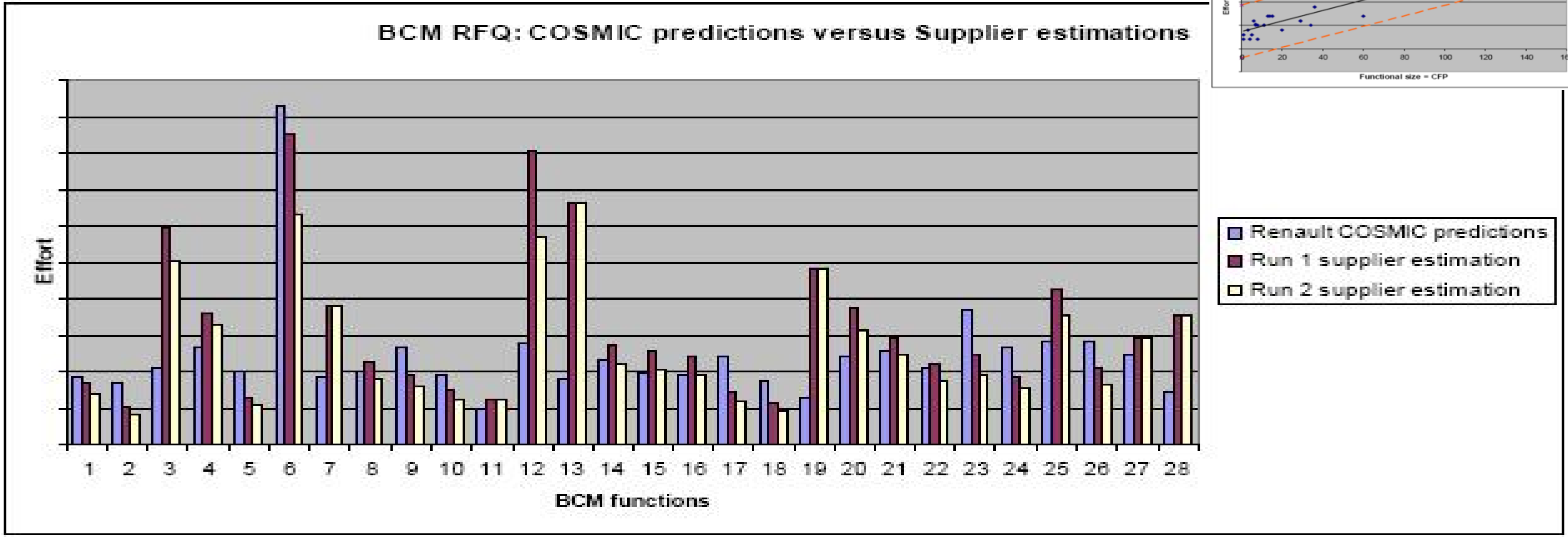


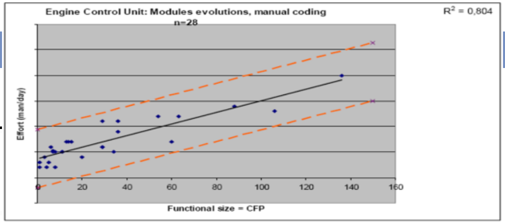
COSMIC

Data from Renault - 2012

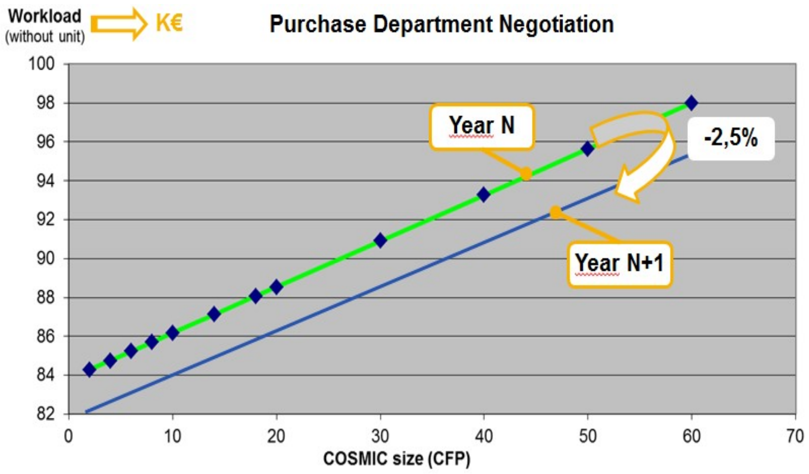


Renault: Estimation & Negociations



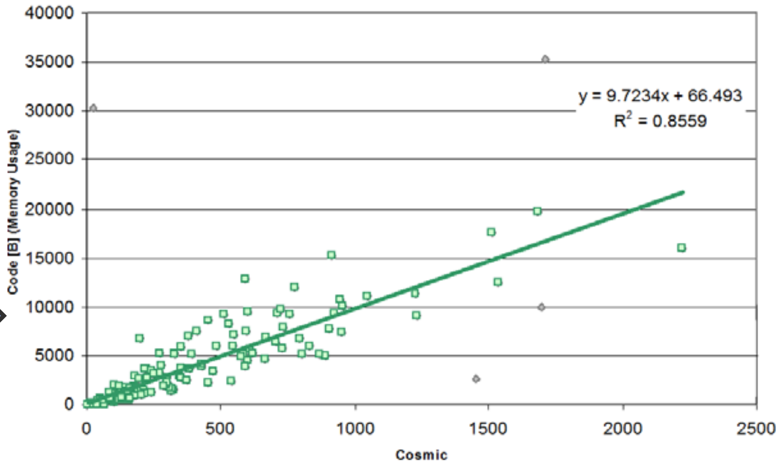


Renault - Remarkable cost estimation accuracy from its ECU software specifications



Cost vs size (CFP)

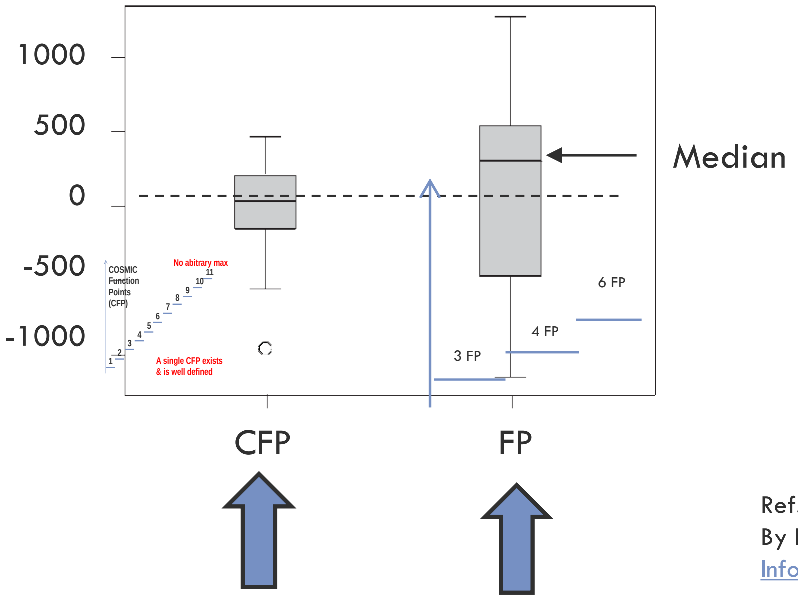
Memory size vs software size (CFP)



COSMIC

Industry Data – Example 2: 25 Web applications

Work-hour Residuals



25 industrial Web applications

Conclusions:

'The results of the … study revealed that COSMIC outperformed Function Points as indicator of development effort by providing significantly better estimations'

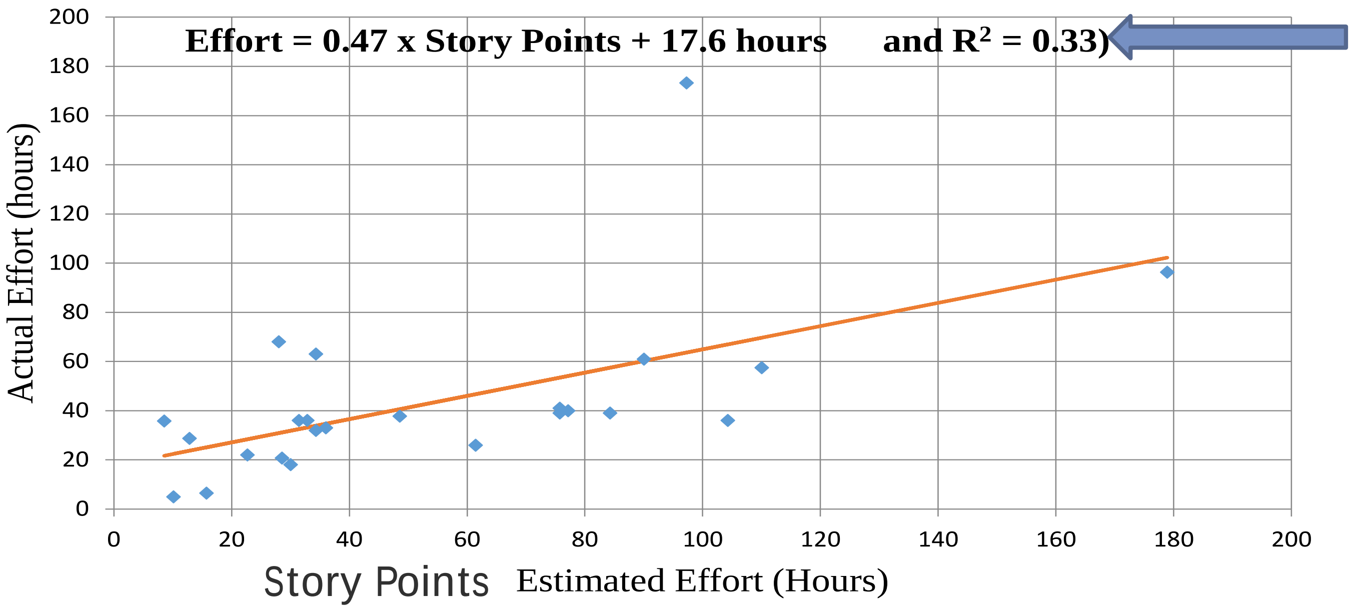
Ref.: 'Web Effort Estimation: Function Point Analysis vs. COSMIC
By Di Martino, Ferrucci, Gravino, Sarro,
Information and Software Technology 72 (2016) 90–109

Industry Data – Example 3: Security & surveillance software systems

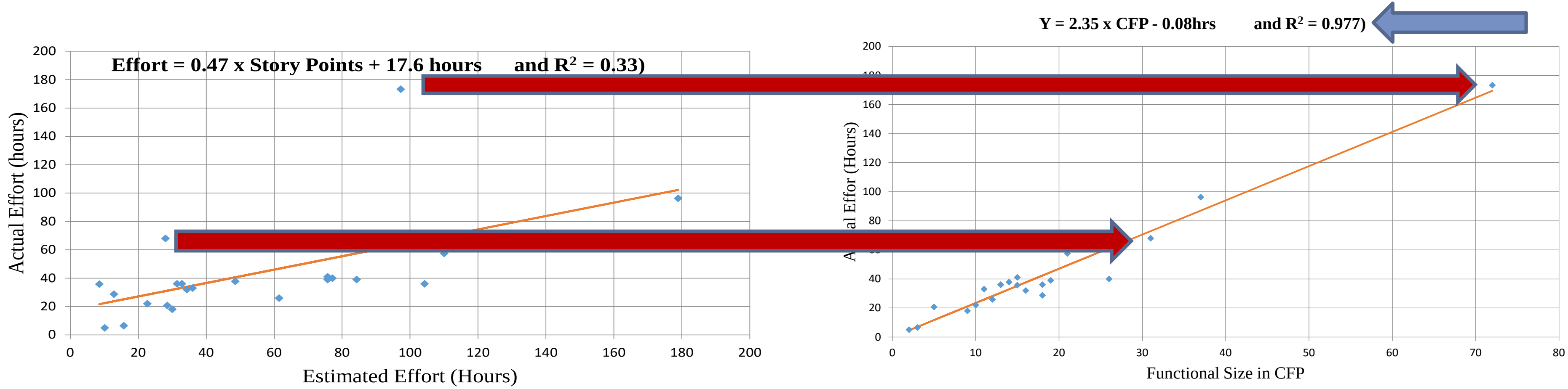
- Scrum method
- Teams estimate tasks within each iteration in Story Points
- Measurements of 24 tasks in 9 iterations
  - Each task estimated in Story Points
  - Task actual effort recorded
  - Each task also measured in CFP

Ref. 'Effort Estimation with Story Points and COSMIC Function Points - An Industry Case Study', *C. Commeyne, A. Abran, R. Djouab.* Obtainable from www.cosmic-sizing.org 'Software Measurement News'. Vol 21, No. 1, 2016

Industry Data – Example 3:
Security & surveillance software systems



Industry Data - Example 3: Security & surveillance software systems



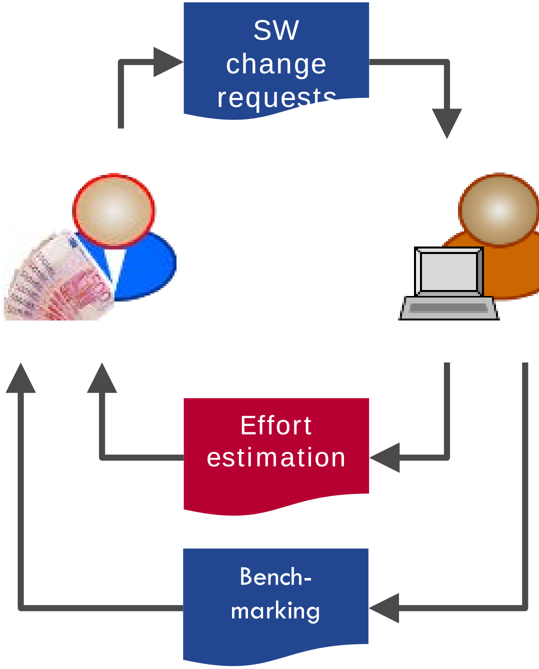
Story Points

COSMIC

Industry Data – Example 4: Vector Consulting Group (Germany) Manufacturing, Engineering, Automotive, ..)

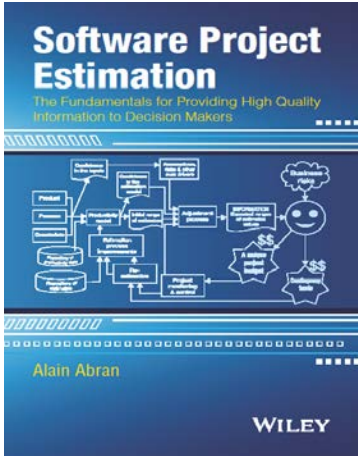
COSMIC Benefits

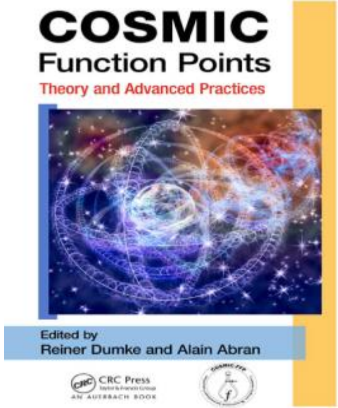
- Agreed model for measuring functional size
  - Solid baseline for benchmarking
- Vector achieved with many clients a precision of 10-20% within one year of building the estimation program:
  - Transparent effort estimations on the basis of functional changes
  - Ad-hoc & fuzzy evaluations and negotiations for single SW changes are reduced
  - Significantly increased efficiency & trust for better collaboration between supplier & customer



Other sources of COSMIC examples

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

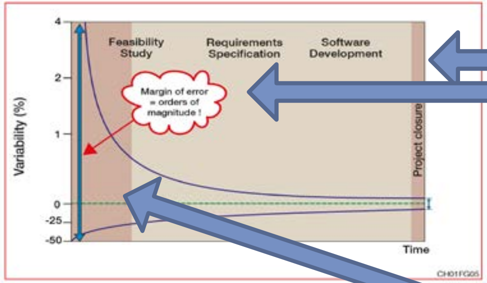




Agenda

1. Software Effort Estimation & Software Size
2. COSMIC: 2nd generation of Function Points
3. Versatility of COSMIC Function Points
4. Contributions of COSMIC to Estimation models
5. Early & Quick COSMIC sizing at estimation time
6. Automation of COSMIC Function Points
7. Summary

Quality of the documentation of a functional process at measurement time

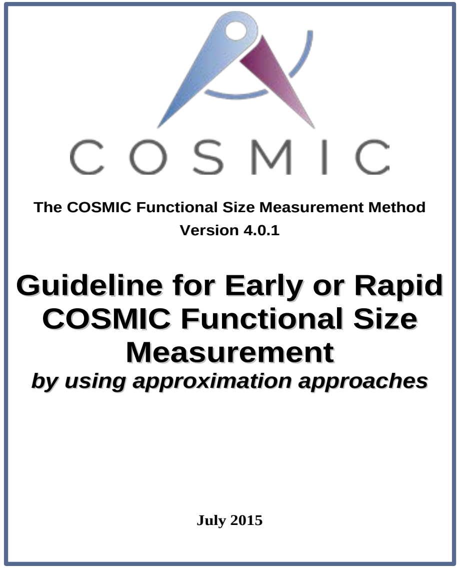


Functional Process Quality Level	Quality of the functional process definition
Completely defined	Functional process and its data movements are completely defined
Documented	Functional process is documented but not in sufficient detail to identify the data movements
Identified	Functional process is listed but no details are given of its data movements
Counted	A count of the functional processes is given, but there are no more details[3]
Implied (A 'known unknown')	The functional process is implied in the actual requirements but is not explicitly mentioned
Not mentioned (An 'unknown unknown')	Existence of the functional processes is completely unknown at present

COSMIC Guidelines for Early or Rapid sizing

Discuss the applicability, reported use, strengths & weaknesses of 8 approximation techniques:

1. Average functional process approximation
2. Fixed size classification approximation
3. Equal size bands approximation
4. Average use case approximation
5. Early & quick COSMIC approximation
6. Easy function points approximation
7. Approximation from informally written texts
8. Approximation using fuzzy logic


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

Example 1: Fixed size intervals

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
…						

COSMIC

Example 2: Equal size bands

Equal size bands from 37 business applications

Band	.Average size of a Functional Process	% of total Functional Size	% of total number of Functional Processes
Small	4.8	25%	40%
Medium	7.7	25%	26%
Large	10.7	25%	19%
Very Large	16.4	25%	15%

Equal size bands from a major component of an avionics system

Band	Average size of a Functional Process	% of total Functional Size	% of total number of Functional Processes
Small	5.5	25%	49%
Medium	10.8	25%	26%
Large	18.1	25%	16%
Very Large	38.8	25%	7%

Example 3: Probability distribution in the Business domain

Classification of the FP	Specification level	CFP (min)	CFP	CFP (max)	Approximate CFP	Probability
Small FP	Little unknown	2 (10%)	3 (75%)	5 (15%)	3.2	>80%
Small FP	Unknown (No FUR)	2 (15%)	4 (50%)	8 (35%)	5.1	<50%
Medium FP	Little unknown	5 (10%)	7 (75%)	10 (15%)	7.25	>80%
Medium FP	Unknown (No FUR)	5 (15%)	8 (50%)	12 (35%)	8.95	<50%
Large FP	Little unknown	8 (10%)	10 (75%)	12 (15%)	10.1	>80%
Large FP	Unknown (No FUR)	8 (15%)	10 (50%)	15 (35%)	11.45	<50%
Complex FP	Little unknown	10 (10%)	15 (75%)	20 (15%)	15.25	>80%
Complex FP	Unknown (No FUR)	10 (15%)	18 (50%)	30 (35%)	21	<50%

COSMIC

Agenda

1. Software Effort Estimation & Software Size
2. COSMIC: 2nd generation of Function Points
3. Versatility of COSMIC Function Points
4. Contributions of COSMIC to Estimation Models
5. Early & Quick COSMIC sizing at estimation time
6. Automation of COSMIC Function Points
7. Summary

COSMIC specifications for Automation with Matlab-Simulink

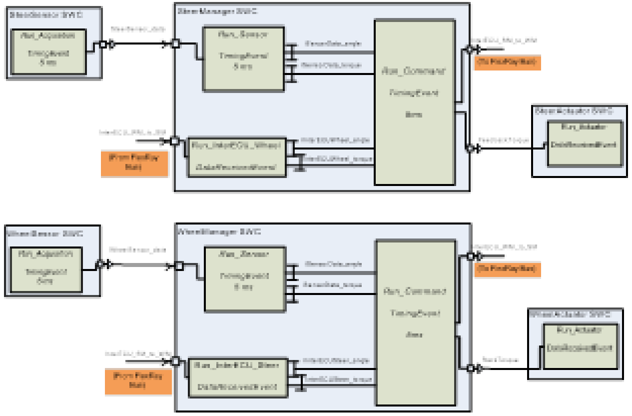
RENAULT	COSMIC Rules	Rev. : v 1.0 Date : 10 May 2012

COSMIC Rules for Embedded Software Requirements Expressed using Simulink®

ABSTRACT

This document presents the rules for measuring with COSMIC (ISO 19761) the functional size of software specifications documented with Matlab-Simulink®.

Map the Graph Notation to COSMIC Model of Software



Ref. H. Soubra, and K. Chaaban, "Functional Size Measurement of Electronic Control Units Software Designed Following the AUTOSAR Standard: A Measurement Guideline Based on the COSMIC ISO 19761 Standard," IWSM-MENSURA Conference, Assisi (Italy), IEEE CS Press, 2012.

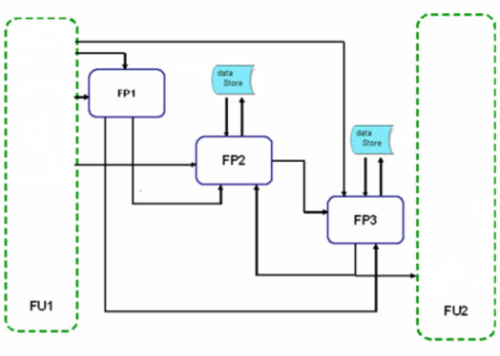


TABLE I.

COSMIC concepts	COSMIC abbreviation	Proposed graphical representation	Proposed graphical description
Functional user	*FU*		Green dashed box
Functional process	FP		Blue box
Data group movement	E/X/W/R		Black arrow
Persistent storage			ISO 5807 stored data symbol in light blue

AUTOMATION ACCURACY REACHED WITH COSMIC

Steer-by-wire case study

Steer-by-Wire Runnable	Functional size obtained by the manual FSM procedure (CFP)	Functional size obtained by the automated FSM procedure (CFP)
Steer_Run_Acquisition	3	3
Steer_Run_Sensor	4	4
Steer_Run_Command	7	7
Steer_InterECU_Wheel	3	3
Steer_Run_Actuator	2	2
Wheel_Run_Acquistion	3	3
Wheel_Run_Sensor	4	4
Wheel_Run_Command	7	7
Wheel_InterECU_Steer	3	3
Wheel _Run_Actuator	2	2
Total	**38**	**38**

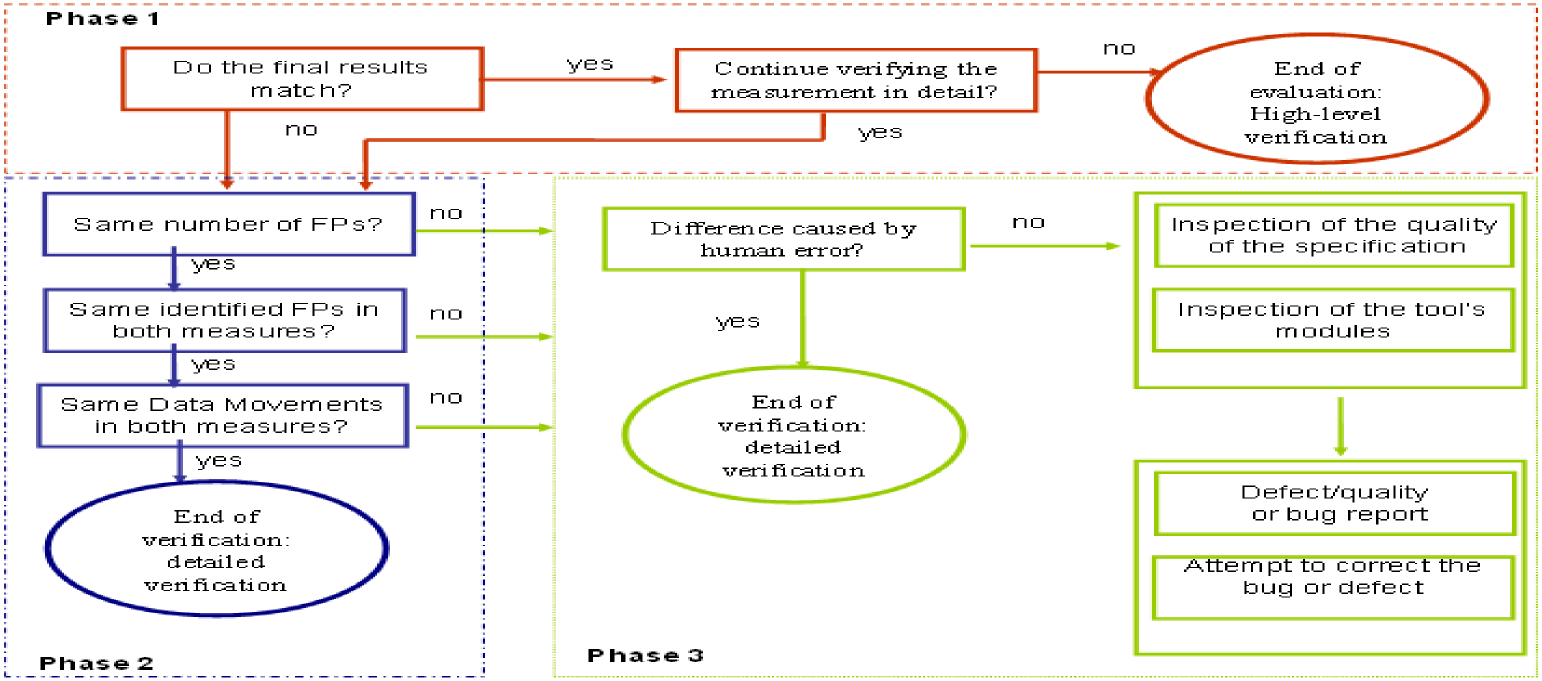
Automation in Industry

Total Number of Models	Total Size obtained manually (CFP)	Total Size obtained using the prototype tool (CFP)	Difference (%)	Accuracy
76 fault-free models	1,729	1,739	Less than 1%	>99%
All 77 models	1,758	1,791	1.8%	>98%

Ref. : Hassan Soubra, Alain Abran, A. R. Cherif,
'Verifying the Accuracy of Automation Tools for the Measurement of Software with COSMIC – ISO 19761 including an AUTOSAR-based Example and a Case Study,'
Joint 24rd International Workshop on Software Measurement & 9th MENSURA Conference, Rotterdam (The Netherlands), Oct. 6-8, 2014, IEEE CS Press, pp. 23-31.

COSMIC

Protocol for Verifying the Accuracy of Automation



COSMIC Automation in SCADE

- Scade: A safety-certified language

https://www.youtube.com/watch?v=gjCvOjaCY88

- https://www.ijerst.com/ijerstadmin/upload/IJEETC_554b274b6329d.pdf

COSMIC

Agenda

1. Software Effort Estimation & Software Size
2. COSMIC: 2nd Generation of Function Points
3. Versatility of COSMIC Function Points
4. Contributions of COSMIC to Estimation Models
5. Early & Quick COSMIC sizing at Estimation Time
6. Automation & Accuracy of COSMIC Function Points
7. Summary

COSMIC

The COSMIC method is used various countries

- COSMIC Measurement Manual standard (11 languages)
- Size of user base is unknown
  - Of known users, 50% are software houses
  - Adopted by Governments (Mexico, Poland, China…)
  - > 30,000 downloads of research & conference papers
- USA: GAO [1], NIST [2] documents
- + 600 certification exam holders (ex. China, India, Mexico, Italy, Poland, Turkey, Brazil)
- Two active forums (on Linkedin CUG, www.cosmic-sizing.org)

1) 'Cost Estimating and Assessment Guide' http://www.gao.gov/new.items/d093sp.pdf , March 2009
2) 'A Rational Foundation for Software Metrology', National Institute for Standards & Technology, NIST IR 8101, January 2016

Summary of benefits

- Free & open
- Fundamental SE Principles: future-proof, stable
- Very wide applicability across software domains & layers
- Proven value for performance measurement & estimating
- ISO standard
- Can be automated with very high accuracy & traceability

Conclusion

Software COST Estimating: critical knowledge for today and tomorrow

Ample industry evidence that COSMIC Function Points allow:

- Meaningfull benchmarking
- Estimation with very low variations (… conditions apply…)
- Automation with high precision

Acknowledgements

The authors wish to acknowledge the efforts of members of the COSMIC Measurement Practices Committee and many others who, over the last 18 years, have contributed to the development and implementation of the COSMIC method

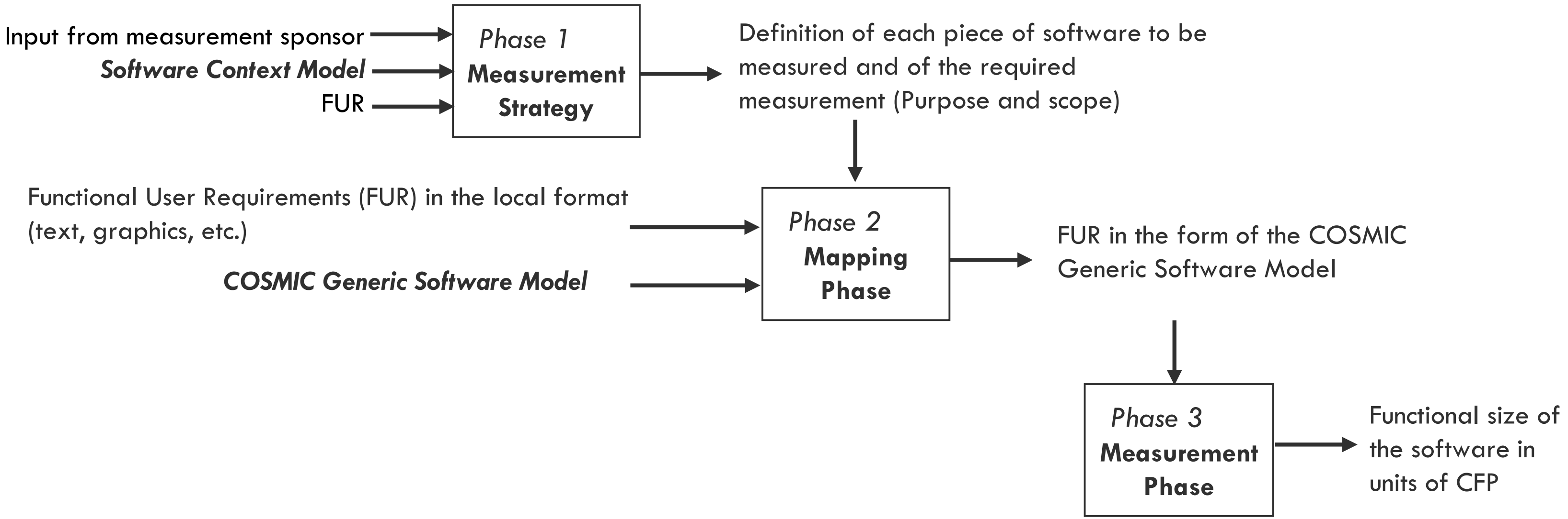
Thank you for your attention

(www.cosmic-sizing.org)

Alain Abran alain.abran@etsmtl.ca
Charles Symons cr.symons@btinternet.com
Christof Ebert christof.ebert@vector.com
Frank Vogelezang frank.Vogelezang@ordina.nl
Hassan Soubra: hassan.soubra@estaca.fr

COSMIC

There is a well-defined Measurement Process

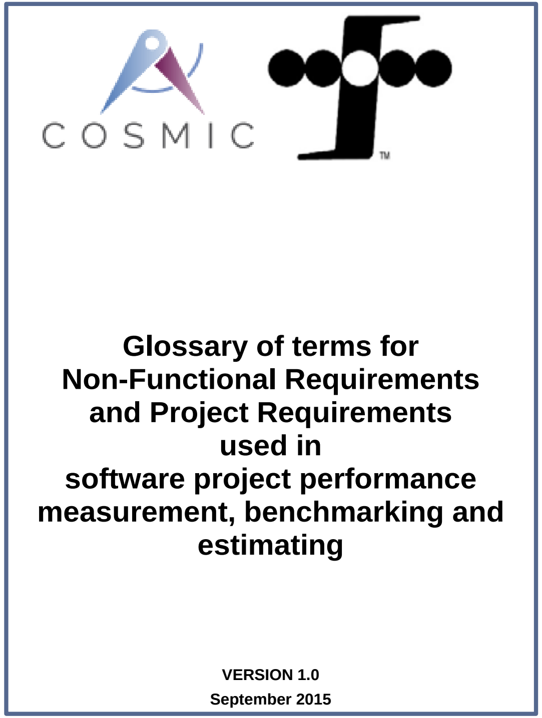


What to do about NFR?

Again, there was no good standard definition of a NFR

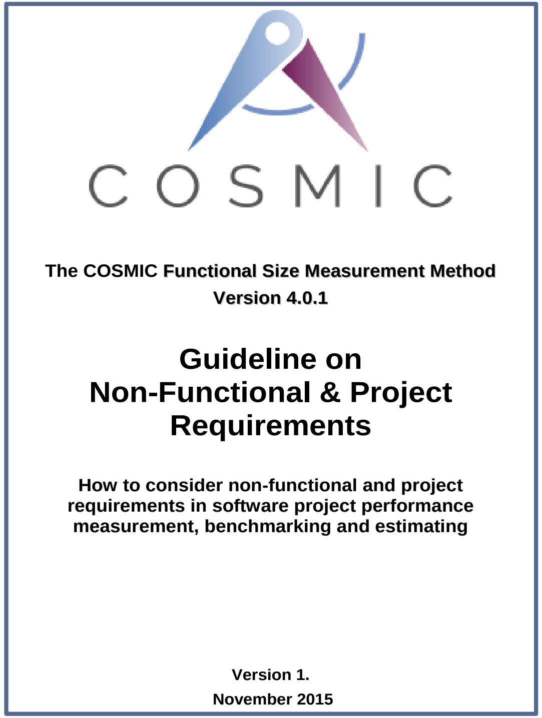
A joint COSMIC/IFPUG effort developed good definitions and a Glossary of NFR and Project Requts.

The COSMIC Guideline advises how to deal with NFR


Glossary of terms for Non-Functional Requirements and Project Requirements used in software project performance measurement, benchmarking and estimating

VERSION 1.0

September 2015


The COSMIC Functional Size Measurement Method

Version 4.0.1

Guideline on Non-Functional & Project Requirements

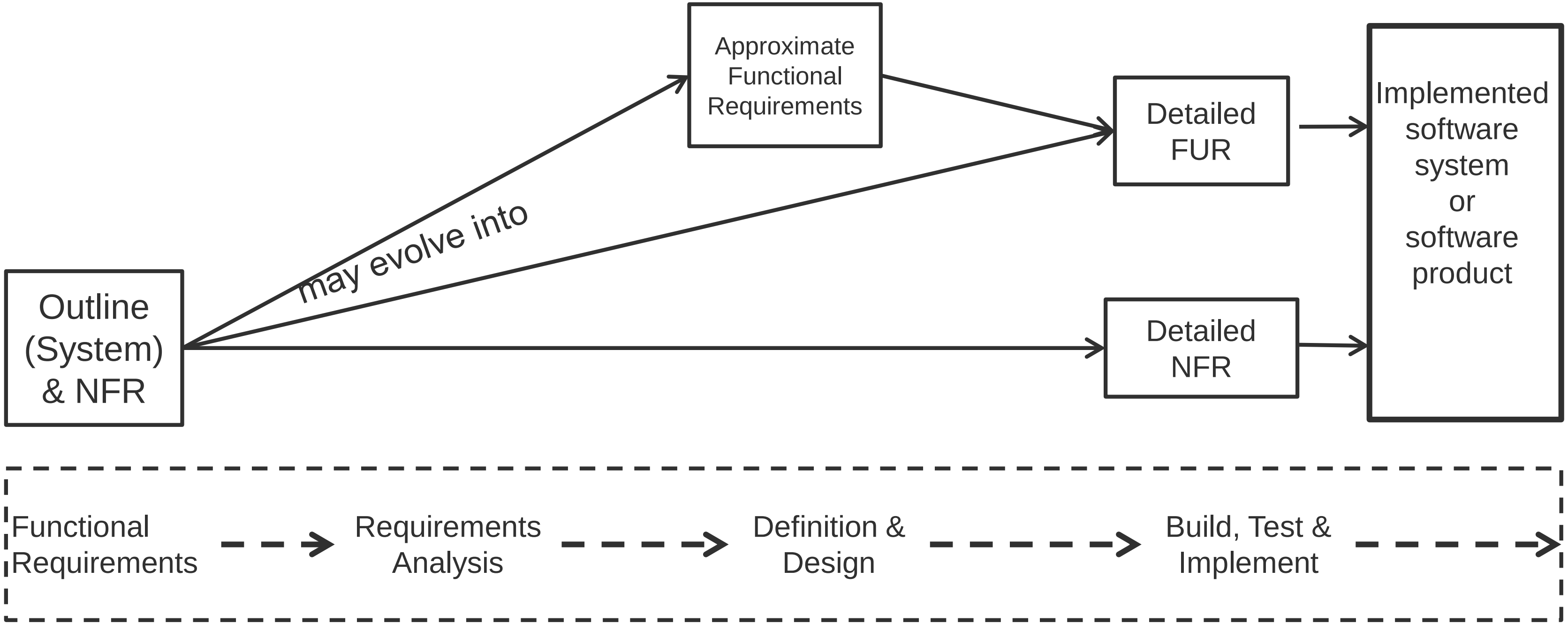
How to consider non-functional and project requirements in software project performance measurement, benchmarking and estimating

Version 1.

November 2015

COSMIC

Abran & Al Sarayreh showed that requirements that appear as NFR may evolve into FUR, that the COSMIC method can measure



Examples of NFR leading to FUR with COSMIC

A Standards-Based Model of System Maintainability Requirements

Khalid T. Al-Sarayreh[*†]
Software Engineering Department, Prince Hussein Bin Abdullah II for Information Technology, Hashemite University (HU), Zarqa, Jordan.

Alain Abran
Software Engineering Department, University of Quebec (ETS), 1100 Notre-Dame Street West, Montréal, Québec H3C 1K3 Canada.

Juan J. Cuadrado-Gallego
Departamento de Ciencias de la Computación, Universidad de Alcalá, 28805 Alcalá de Henares, Madrid, Spain

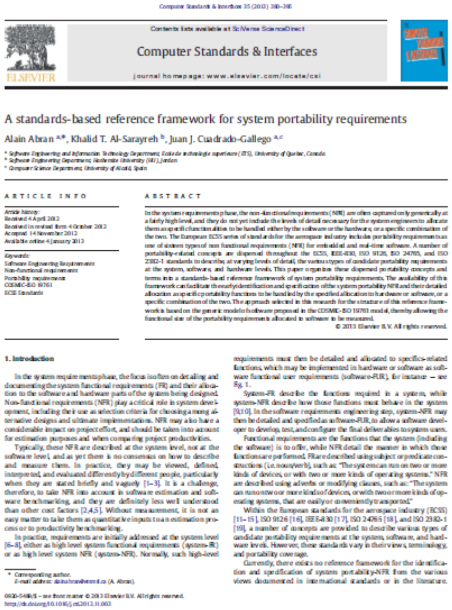
1. SUMMARY

The non functional requirements are often captured only generically at a fairly high level, and they do not include the levels of detail necessary at this stage for the system engineers to allocate them as specific functionalities to be handled either by the software or the hardware, or a specific combination of the two. The European ECSS series of standards for the aerospace industry includes maintainability requirements as one of sixteen types of non functional requirement (NFR) for embedded and real-time software. A number of maintainability-related concepts are dispersed throughout the ECSS, ISO 9126, and IEEE standards to describe, at varying levels of detail, the various types of candidate maintainability requirements at the system, software, and hardware levels. This paper organizes these dispersed maintainability concepts into a standards-based reference model of system maintainability requirements. The availability of this reference model can facilitate the early identification of the system maintainability-NFR and their detailed allocation as specific maintainability functions to be handled by the specified allocation to hardware or software, or a specific combination of the two. In the absence of such a reference model, these NFR requirements are typically handled in practice much later on in the software development life cycle, when, at system testing time, users and developers find out that a number of maintainability requirements have been overlooked and additional effort has to be expended to implement them. The approach adopted in this research for the structure of this reference NFR model is based on the generic model of software functional requirements proposed in the COSMIC – ISO 19761 model, thereby allowing the functional size of such maintainability requirements allocated to software to be measured. Copyright © 2011 John Wiley & Sons, Ltd.

KEYWORDS: Software Engineering, Non Functional Requirements – NFR, Maintainability Requirements, ECSS International Standards, Maintainability Measurement, Functional size, COSMIC – ISO 19761.

* Correspondence to: Khalid T. Al-Sarayreh, Software Engineering Department, Hashemite University (HU), 1100 Zarqa, Jordan. †E-mail: khalidt@hu.edu.jo

1



Computer Standards & Interfaces 35 (2013) 380–395
Contents lists available at SciVerse ScienceDirect
Computer Standards & Interfaces
journal homepage: www.elsevier.com/locate/csi

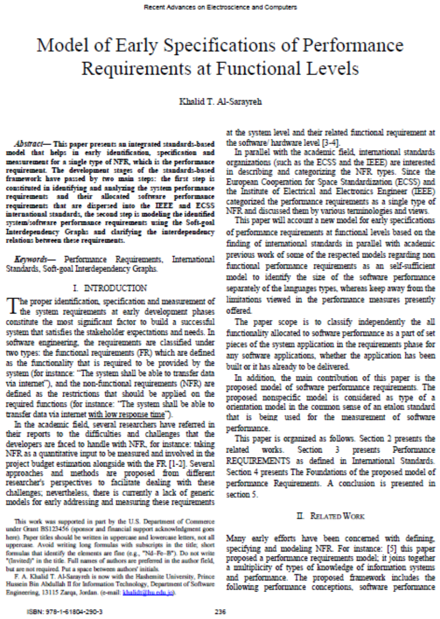
A standards-based reference framework for system portability requirements

Alain Abran [a,*], Khalid T. Al-Sarayreh [b], Juan J. Cuadrado-Gallego [a,c]

ARTICLE INFO

Article history:
Received 4 April 2012
Received in revised form 4 October 2012
Accepted 14 November 2012

ABSTRACT



Recent Advances on Electroscience and Computers

Model of Early Specifications of Performance Requirements at Functional Levels

Khalid T. Al-Sarayreh

Abstract— This paper presents an integrated standards-based model that helps in early identification, specification and measurement for a single type of NFR, which is the performance requirement. The development stages of the standards-based framework have passed by two main steps: the first step is constituted in identifying and analyzing the system performance requirements and their allocated software performance requirements that are dispersed into the IEEE and ECSS international standards, the second step is modeling the identified system/software performance requirements using the Soft-goal Interdependency Graphs and clarifying the interdependency relations between these requirements.

Keywords— Performance Requirements, International Standards, Soft-goal Interdependency Graphs.

I. INTRODUCTION

The proper identification, specification and measurement of the system requirements at early development phases constitute the most significant factor to build a successful ...

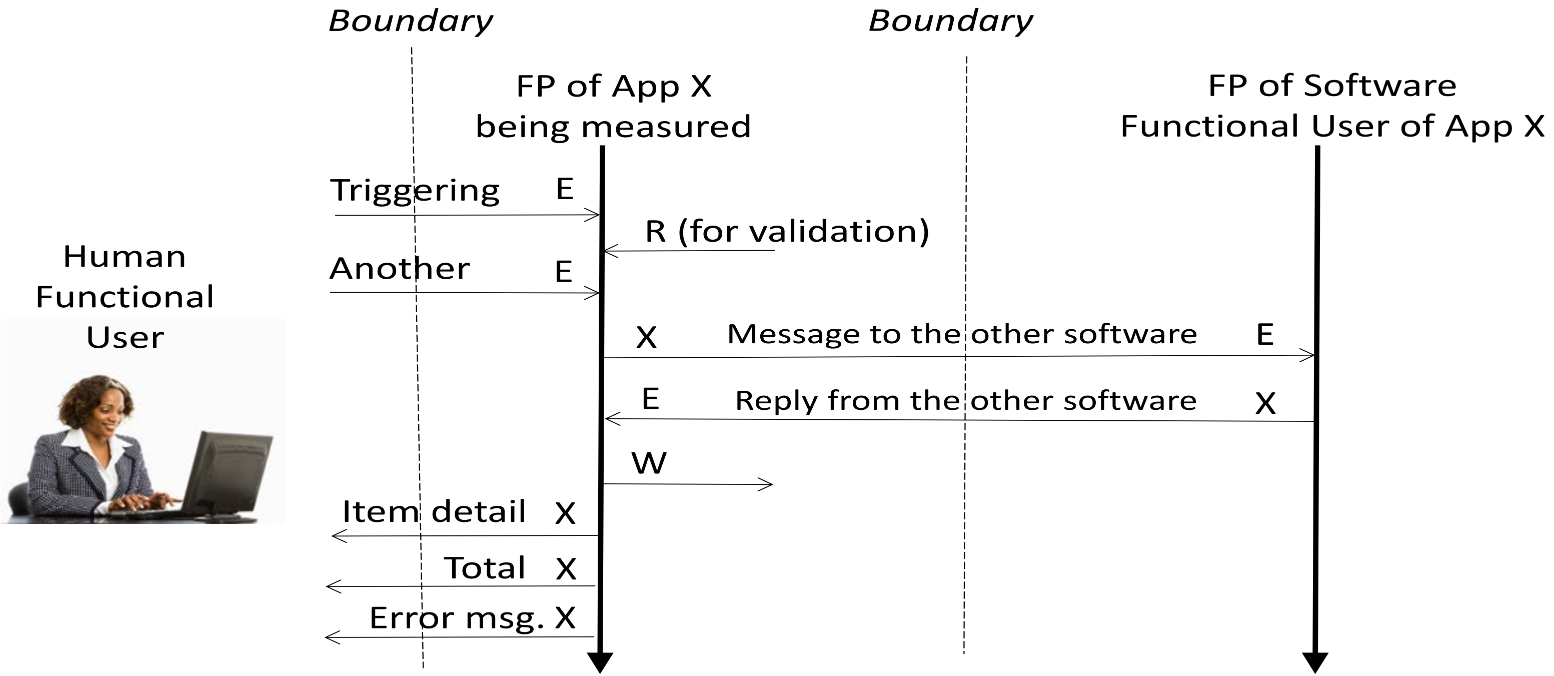
... at the system level and their related functional requirement at the software/ hardware level [3-4].

In parallel with the academic field, international standards organizations (such as the ECSS and the IEEE) are interested in describing and categorizing the NFR types. Since the European Cooperation for Space Standardization (ECSS) and the Institute of Electrical and Electronics Engineer (IEEE) categorized the performance requirements as a single type of NFR and discussed them by various terminologies and views.

This paper will account a new model for early specifications of performance requirements at functional levels based on the finding of international standards in parallel with academic previous work of some of the respected models regarding non functional performance requirements as an self-sufficient model to identify the size of the software performance separately of the languages types, whereas keep away from the limitations viewed in the performance measures presently offered.

COSMIC

Example 2: with a Message Sequence Diagram



Size = 9 CFP

Guidelines for Practitioners

A Guideline describing a range of Approximate Sizing methods

Size/Cost estimates are usually needed before the FUR have been defined in detail

A Guideline on **'Assuring the accuracy of COSMIC measurements'**

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 Common Software Measurement International Consortium

The COSMIC Functional Size Measurement Method
Version 3.0.1

Guideline for assuring the accuracy of measurements

VERSION 0.93
February 2011

COSMIC

1st Generation of Function Points: Weights

FTR's	DATA ELEMENTS		
	1-4	5-15	> 15
0-1	Low	Low	Ave
2	Low	Ave	High
3 or more	Ave	High	High

Inputs - Matrix

FTR's	DATA ELEMENTS		
	1-5	6-19	> 19
0-1	Low	Low	Ave
2-3	Low	Ave	High
> 3	Ave	High	High

Output & Enquiries – Shared Matrix

Rating	VALUES		
	EO	EQ	EI
Low	4	3	3
Average	5	4	4
High	7	6	6

Transactions: weights in FP

RET's	DATA ELEMENTS		
	1-19	20 - 50	> 50
1	Low	Low	Ave
2-5	Low	Ave	High
> 5	Ave	High	High

Files (internal & external) Matrix

Rating	Values	
	ILF	EIF
Low	7	5
Average	10	7
High	15	10

Files: weights in FP