



MEASUREMENT OF SOFTWARE SIZE: CONTRIBUTIONS OF COSMIC TO ESTIMATION IMPROVEMENTS

Alain Abran

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

**ICEAA International Training Week
October 17-20, 2016, Bristol (UK)**

Presenter background: Alain Abran

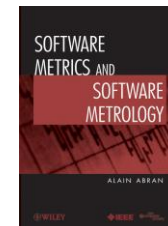
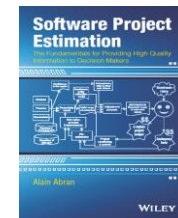
20 years



- Development
- Maintenance
- Process Improvement

+ 35 PhD

20 years




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



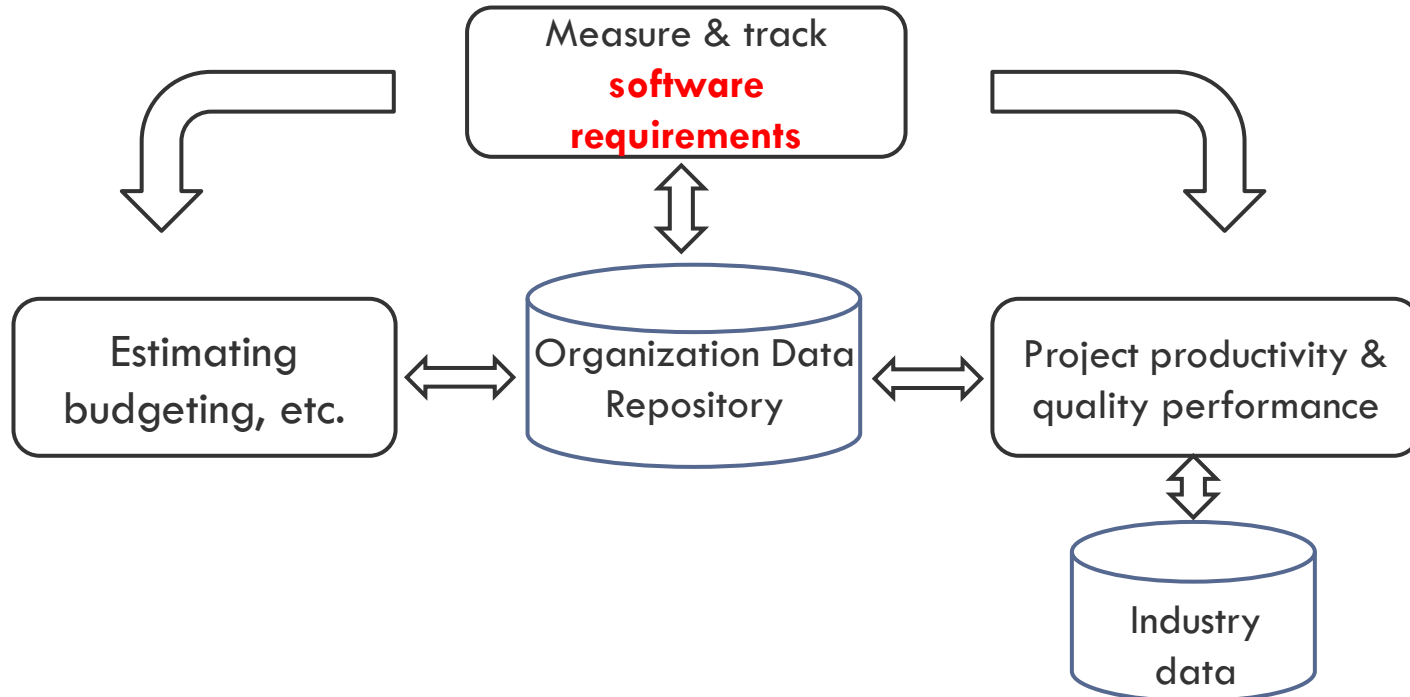
Agenda

3

- 
- **Background to Functional Size Measurement (FSM) methods**
 - **COSMIC Method – Key features of ISO 19761**
 - **Measurement Guidelines**
 - **‘The proof of the pudding is in the eating’: Good Estimation**
 - **Automation of COSMIC measurement**
 - **Conclusions**

Objective: we want to use performance data for estimating future projects

4



Software Sizing Options

5

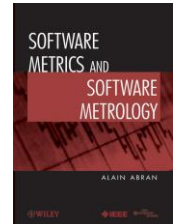
Sizing method options:

- 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 (Planning Poker):
 - X Entirely Subjective & Benchmarking impossible:
 - **unaccountability**

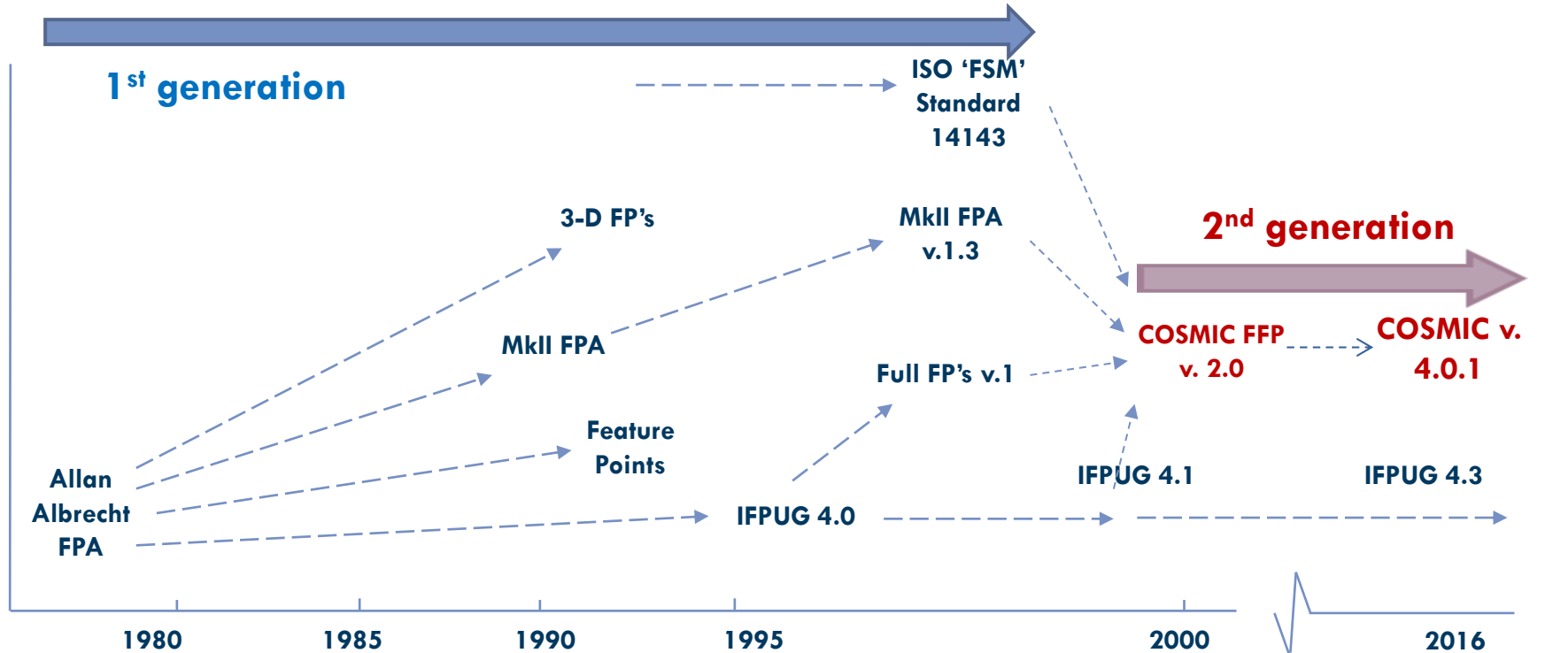
- Functional size
 - ✓ International standard methods
 - ✓ (Function Points): ✓ Technology-independent





1st & 2nd generation of Function Points Methods

6





Agenda

7

- **Background to Functional Size Measurement (FSM) methods**
- ➔ ■ **COSMIC FSM Method – Key features of ISO 19761**
- **Measurement Guidelines**
- **‘The proof of the pudding is in the eating’: Good Estimation**
- **Automation of COSMIC measurement**
- **Conclusions**



COSMIC view of software

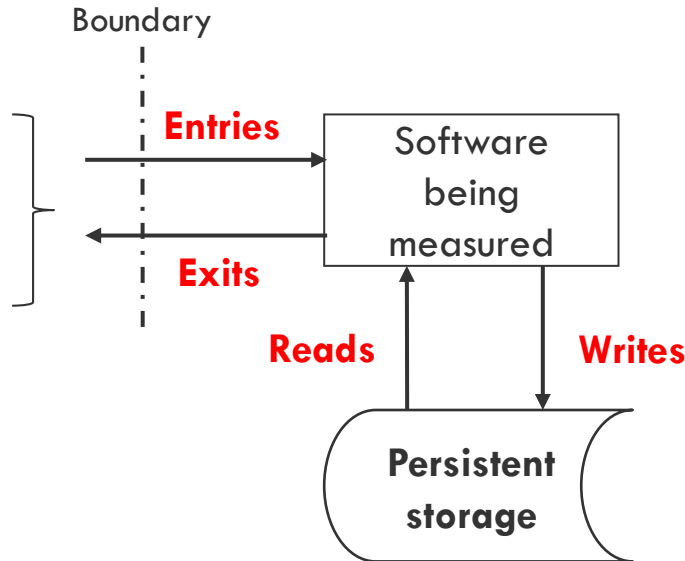
8

- What is common across all software, in different types of software, whether very small or extremely large?

All software does this!

Functional Users

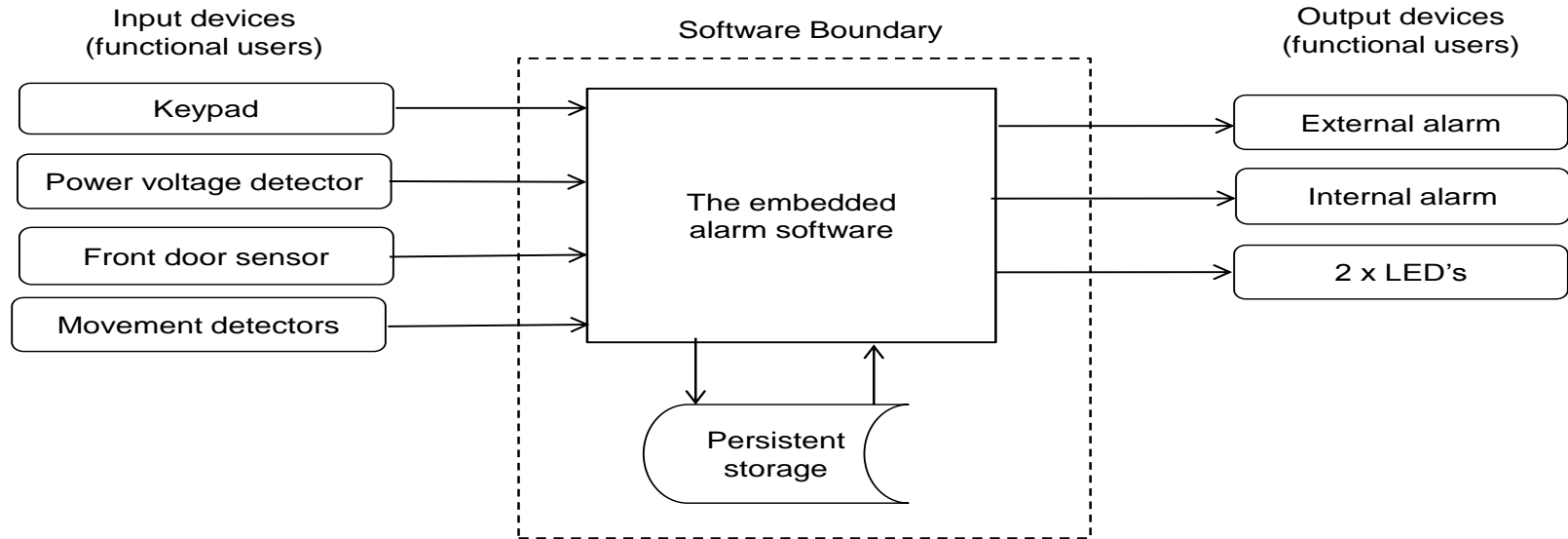
- Hardware devices,
- Other software or
- Humans



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

Example: Intruder Alarm System

10



Intruder Alarm System

Functional process: Possible intruder detected.

Triggering event: Door opens whilst alarm system is activated.

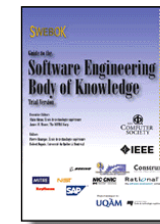
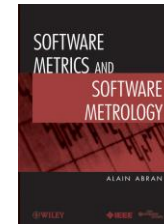
Data Movement	Functional User	Data Group
Entry	Front-door sensor	'Door open' message (triggering Entry)
Read	- / Occupant	PIN (from persistent storage)
Exit	Green LED	Switch 'off' command
Exit	Red LED	Switch 'on' command
Exit	Internal siren	Start noise command
Entry	Keypad	PIN (If the wrong code is entered, 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)

COSMIC Method

12

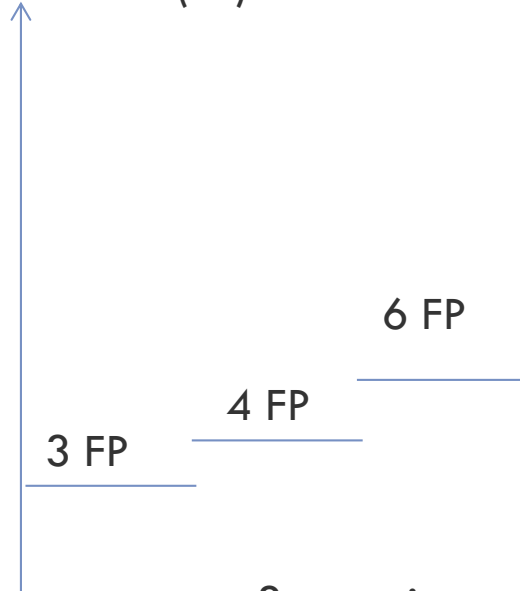
- Designed by an international group of software measurement experts
 - COSMIC: Common Software Measurement International Consortium
- To measure the Functional User Requirements of:
 - Business application
 - Real-time
 - Infrastructure software
 - Various other types of software
 - Hybrids of these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



1st Generation of Function Points: Step Functions

13

Function Points (FP)



Key limitations:

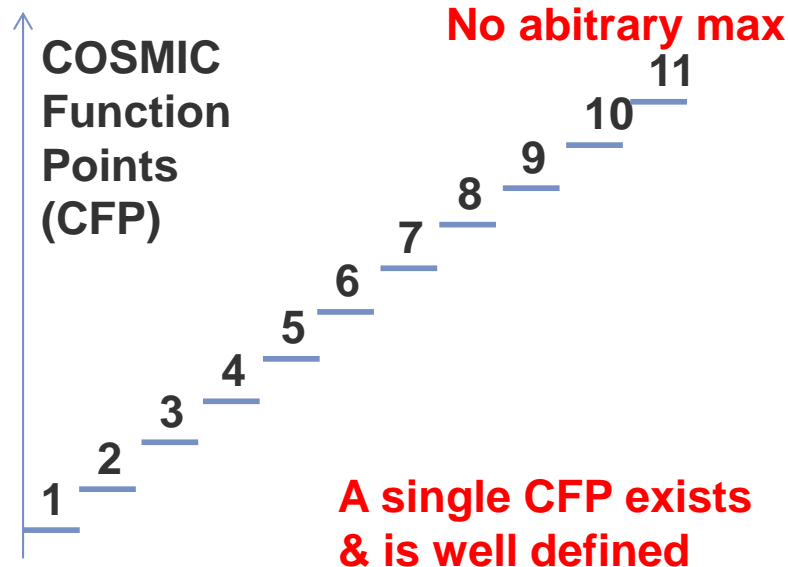
- Only 3 values
- Limited ranges (min,max)
- No single measurement unit of 1 FP!

3-step size range for the IFPUG External Input Transactions



2nd Generation with COSMIC

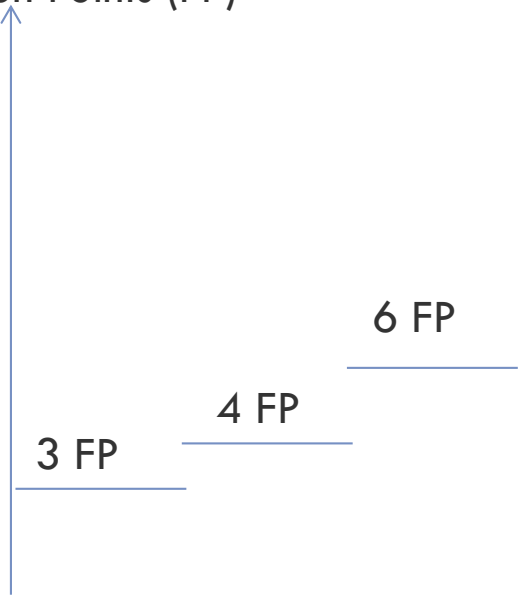
14



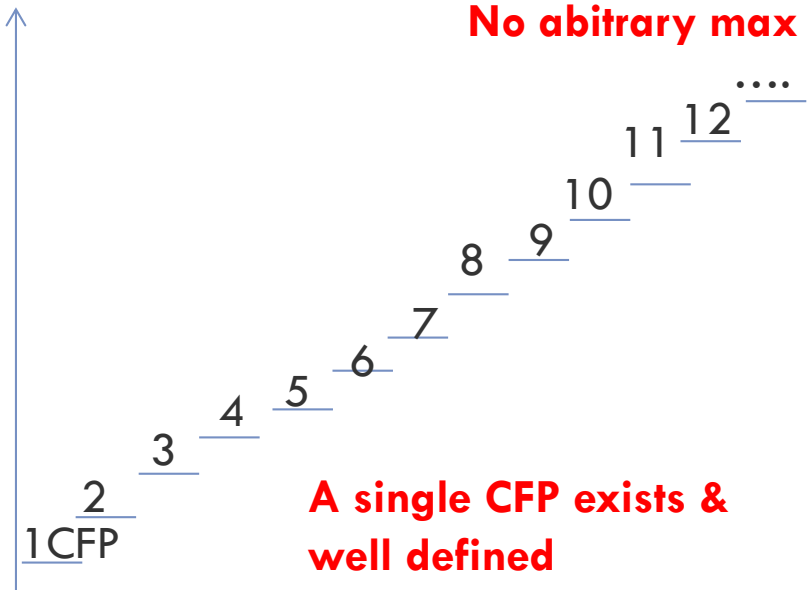


1st and 2nd Generations of FSM

Function Points (FP)



COSMIC Function Points - CFP



No arbitrary max

A single CFP exists & well defined



COSMIC sizes are measured on a true ratio scale

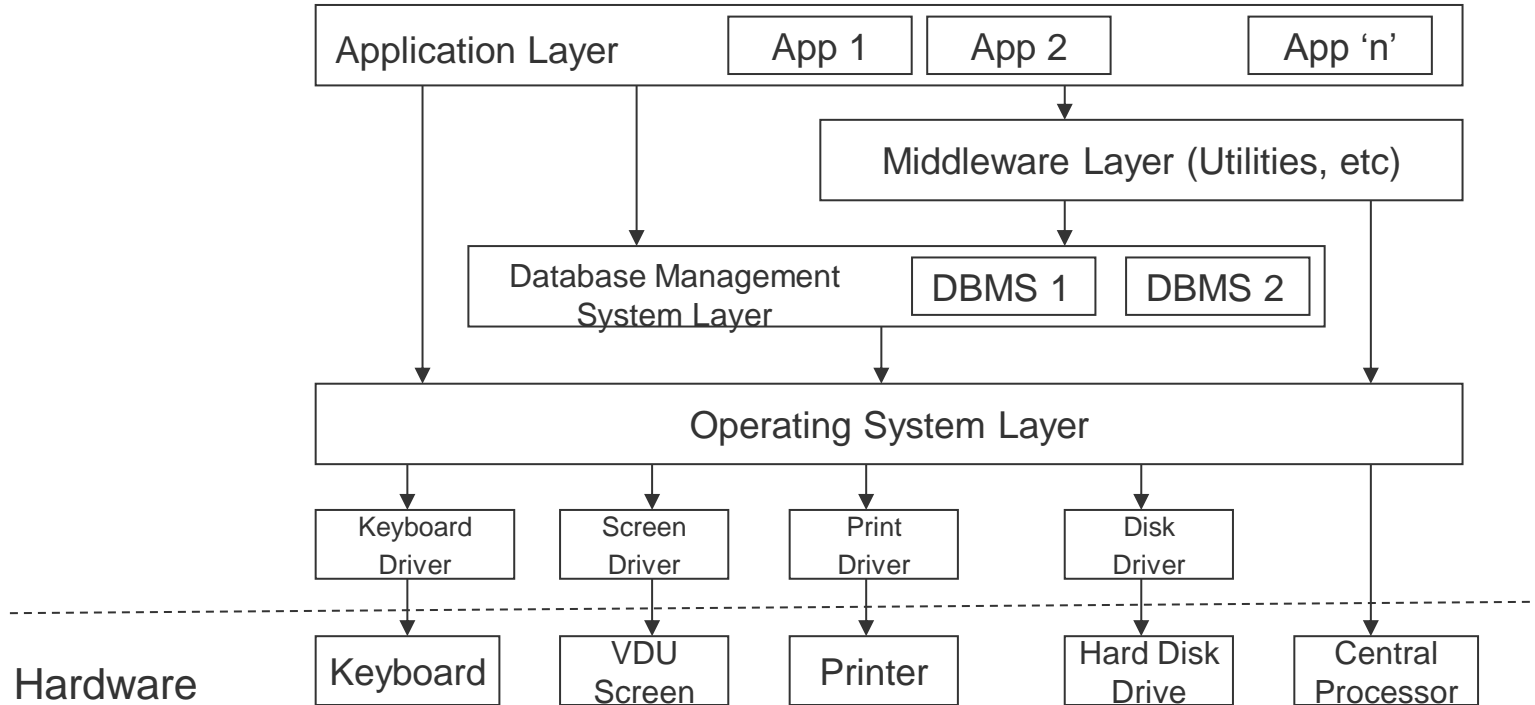
16

- **There is no upper limit to the size of a functional process**
- **Largest observed functional processes?**
 - **In avionics >100 CFP**
- **The size of the smallest change to an existing functional process is 1 CFP**
- **Open, freely available (via www.cosmic-sizing.org)**



COSMIC - at any level of software requirements

17



Agenda

18

- **Background to Functional Size Measurement (FSM) methods**
- **COSMIC FSM Method – ISO 19761**
- ➔ ■ **Measurement Guidelines**
- **‘The proof of the pudding is in the eating’: Good Estimation**
- **Automation of COSMIC measurement**
- **Conclusions**



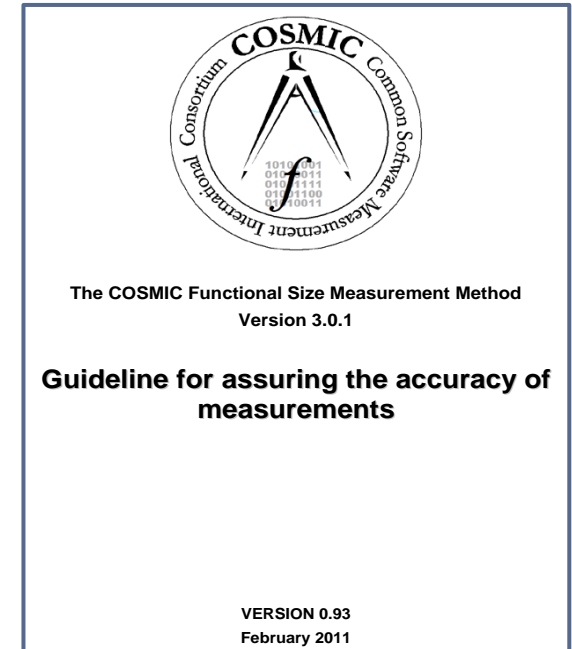
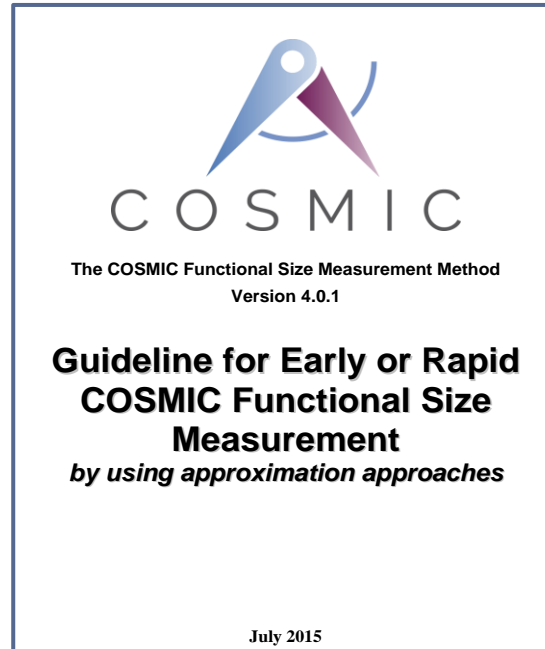
Recent Guidelines for Practitioners

19

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'





Guidelines by Application Domains

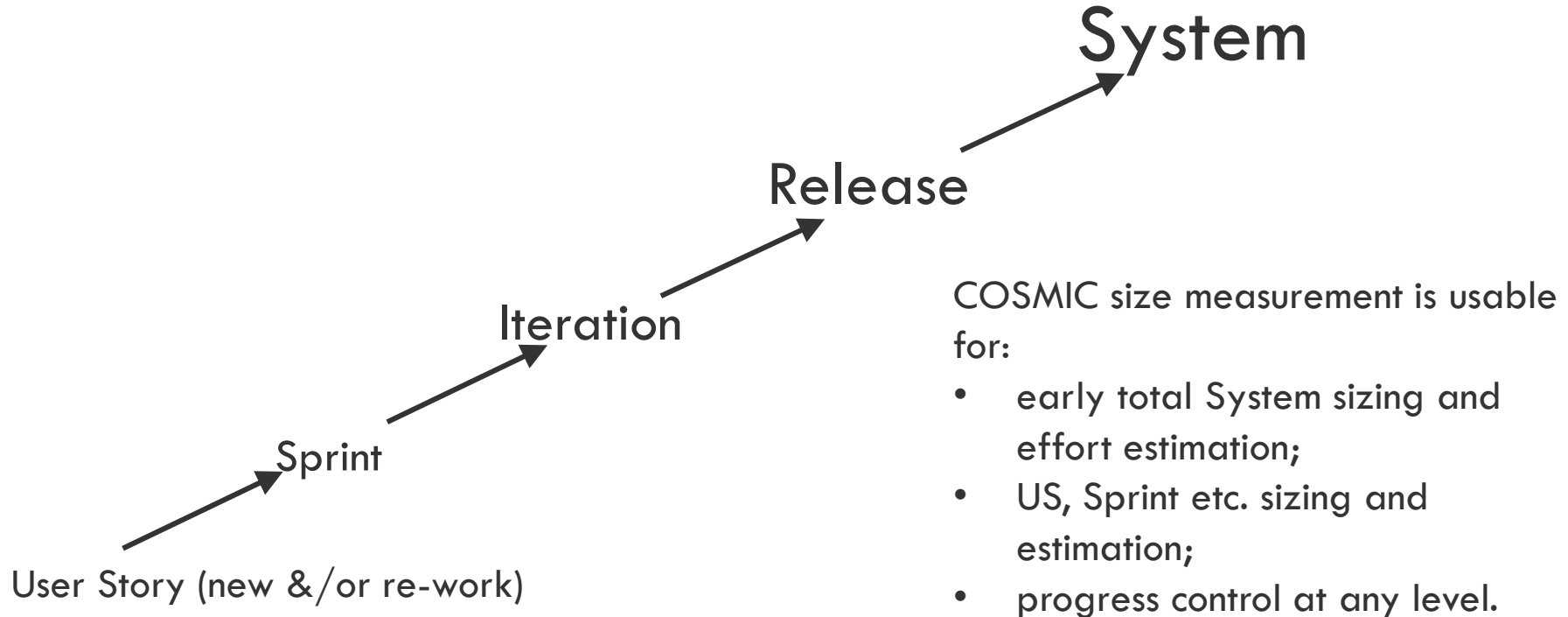
20

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

and for **Agile Developments**



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





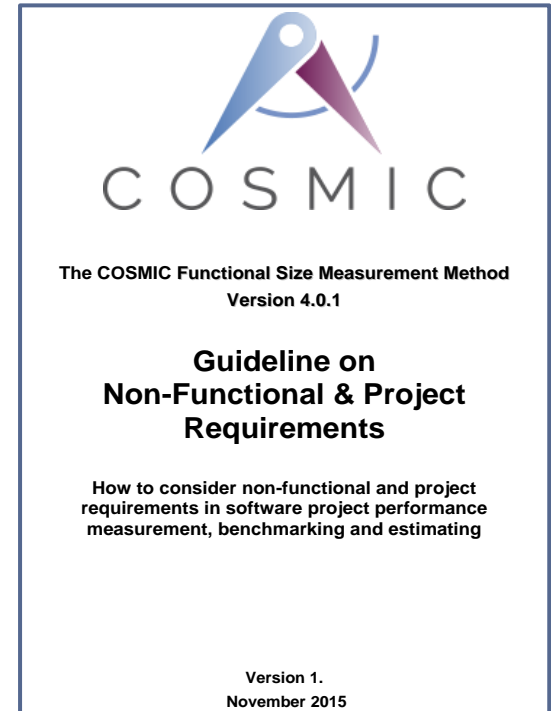
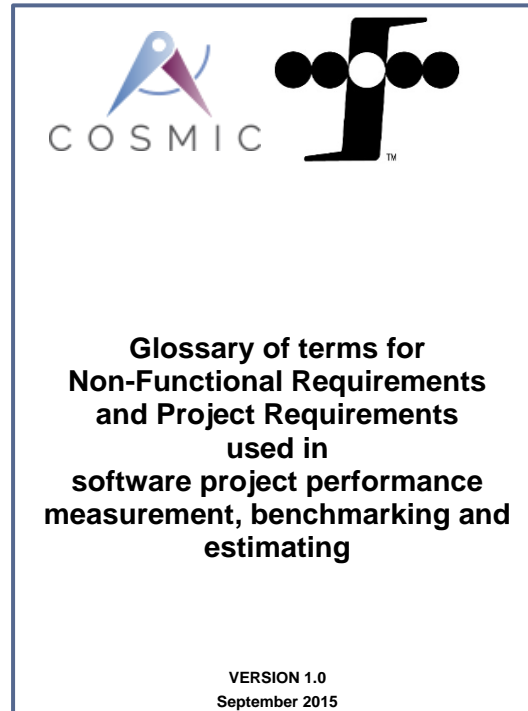
What to do about NFR?

22

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 Requets.

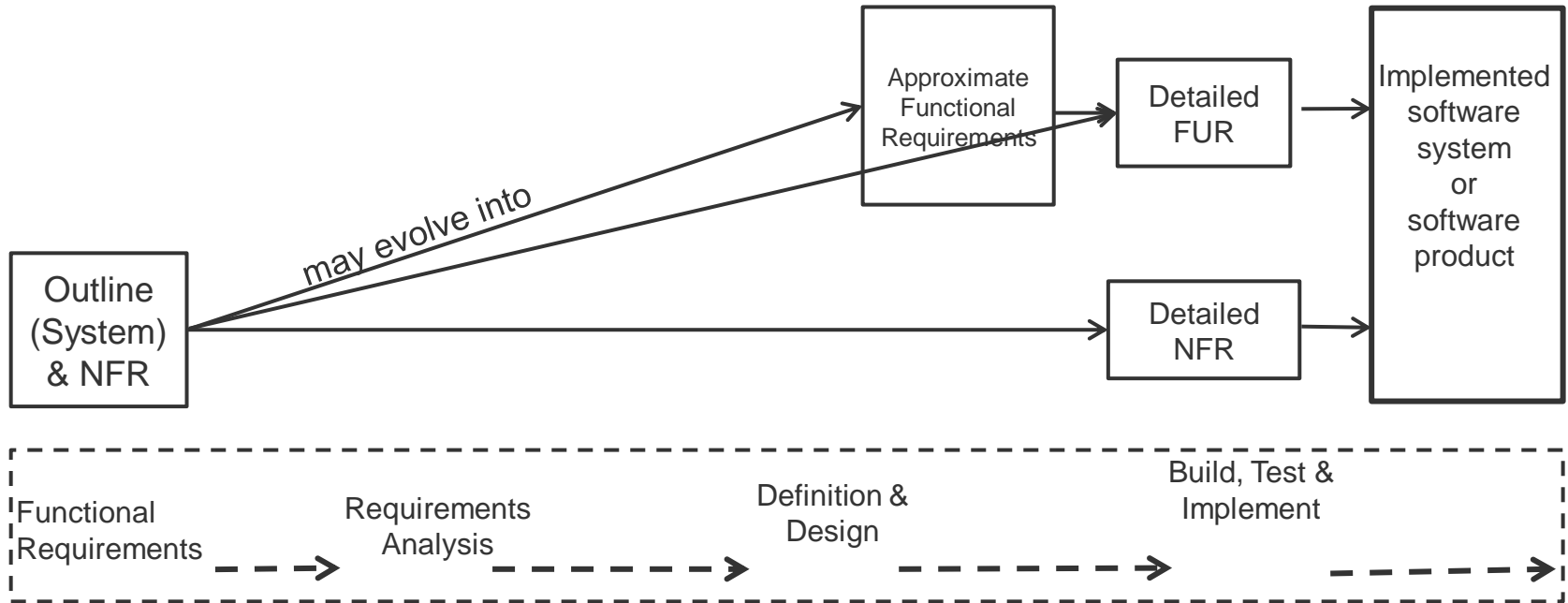
The COSMIC Guideline advises how to deal with NFR





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

23





Examples of NFR leading to FUR with COSMIC

A Standards-Based Model of System Maintainability Requirements

Khalid T. Al-Sarayah^{1*}
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, Montreal, Quebec H3C 1K3 Canada.

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



A standards-based reference framework for system portability requirements

Alain Abran^{1*}, Khalid T. Al-Sarayah¹, Juan J. Cuadrado-Gallego²

¹Software Engineering Department, University of Quebec (ETS), 1100 Notre-Dame Street West, Montreal, Quebec H3C 1K3 Canada.
²Departamento de Ciencias de la Computación, Universidad de Alcalá, 28805 Alcalá de Henares, Madrid, Spain

ARTICLE INFO
ABSTRACT
Keywords
Non-functional requirements
Portability requirements
COSMIC-ISO 19761
ISO 19761

Portability requirements have, for the last three decades, been mostly captured by generically or fairly high-level and they do not yet include the level of detail necessary 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 a number of generic top-level functional requirements (NFR) for embedded and real-time software. A number of portability-related concepts are dispersed throughout the ECSS, ISO 9126, ISO 9246, ISO 24760, and ISO 23252 standards describing, at one level, details of the system portability requirements. The availability of the framework can facilitate the identification and allocation of the system portability NFRs that describe allocation and portability functions to be handled by the specified architecture software or hardware, or a specific combination of the two. This approach is used in the research for the creation of the reference framework for handling the generic models of these proposed in the COSMIC-ISO 19761 model. Finally, additional functional and/or portability requirements allocated to software is also discussed.

1. Introduction

The system require ment phrase, the fact is often on detailing and decomposing the system functional requirements (FR) and their allocation to the software and hardware parts of the system being designed. Non-functional requirements (NFR) play a critical role in system development, including their use as selection criteria for choosing among alternative designs and software implementations. NFR may only have a considerable impact on project efforts, and should be taken into account for optimization purposes and when comparing project products. Typically, these NFR are considered at the system level, not at the software level, and as yet there is no consensus on how to describe and measure them. In practice, they may be specified, defined, interpreted, and evaluated differently by different people, particularly when they are stated briefly and vaguely [1-5]. It is a challenge, therefore, to take NFR into account in software estimation and software development, and they are definitely less well understood than other cost factors [2,4,5]. Without measurement, it is not an easy matter to take them into account to improve cost estimation process or to predictively benchmarking.

requirements must then be identified and abstracted to specific-related functions which may be implemented in hardware or software or software functional user requirements (software-NFR). For instance – see Fig. 1 – System-NFR describe the functions required in a system, while system-NFR describe how these functions must behave in the system [10]. In the software requirements engineering step, system-NFR may then be handled and specified in terms of software-NFR to allow a software developer to develop, test, and debug the final deliverable system users.

* Corresponding author.
E-mail address: alksarayah@uqam.ca (A. Abran).
URL: <http://www.iceaaonline.com/bristol2016>
URL: alksarayah@uqam.ca
URL: juan.j.cuadrado@uca.es

Model of Early Specifications of Performance Requirements at Functional Levels

Khalid T. Al-Sarayah

Abstract—This paper proposes 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 steps of the standards-based framework have passed by two main steps: the first step is concerned in identifying and making 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 Sub-goal Interdependency Graph and clarifying the interdependency relations between their requirements.

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

1. INTRODUCTION

The proper identification, specification and measurement of the system requirements at early development phases constitute the most significant factor to build a successful system that satisfies the stakeholder expectations and needs. In software engineering, the requirements are classified under two types: the functional requirements (FR) which are defined as the functionality that is required to be provided by the system (for instance: “The system shall be able to transfer data via internet”), and the non-functional requirements (NFR) are defined as the restrictions that should be applied on the required functions (for instance: “The system shall be able to transfer data via internet with low response time”).

In the academic field, several researchers have referred in their reports to the difficulties and challenges that the developers are faced to handle with NFR, for instance, taking NFR is a quantitative input to be measured and involved in the project budget estimation alongside with the FR [1-3]. Several approaches and methods are proposed in the recent researcher’s perspectives to facilitate dealing with these challenges; nevertheless, there is currently a lack of generic models for early addressing and measuring these requirements

at the system level and their related functional requirement at the software/hardware level [1-4]. In parallel with the academic field, international standards organizations (such as the ECSS and the IEEE) are interested in describing and capturing the NFR types. Since the European Cooperation for Space Standardization (ECSS) and the Institute of Electrical and Electronics Engineers (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 reported models regarding non-functional performance requirements as an self-sufficient model to identify the size of the software performance requirements of the language types, which keep away from the hazardous viewed in the performance measures presently offered.

The paper scope is to classify independently the all functionally allocated to software performance as a part of set pieces of the system application in the requirement phase for any software application, whether the application has been built or is already to be delivered. In addition, the main contribution of this paper is the proposed model of software performance requirements. The proposed specific model is considered as type of a orientation model in the common sense of an evolve standard that to be used to help for the measurement of software performance.

This paper is organized as follows: Section 2 presents the related work. Section 3 presents Performance Requirements as defined in International Standards. Section 4 presents the Foundations of the proposed model of Performance Requirements. A conclusion is presented in section 5.

II. RELATED WORK

Many early efforts have been concerned with defining, specifying and modeling NFR. For instance [5] this paper proposed a performance requirements model. It joins together a multiplicity of types of knowledge of information system and performance. The proposed framework includes the following performance conception, software performance

This work was supported by the U.S. Department of Commerce under Grant H5123456 system and financial support acknowledgment goes here. These titles should be written in uppercase and lowercase letters, and all uppercase. Avoid writing long formulae with subscripts in the title; short formulae should be italicized and numbered (e.g., “ 100×10^6 Hz”) but not with “(Hz)”. In the title, full names of authors are preferred to the author field, but are not used in the title.

* Khalid T. Al-Sarayah is now with the Hashemite University, Prince Hussein Bin Abdullah II for Information Technology, Department of Software Engineering, 13110 Zarqa, Jordan (e-mail: alksarayah@uqam.ca).

* Correspondence to: Khalid T. Al-Sarayah, Software Engineering Department, Hashemite University (HU), 1100 Zarqa, Jordan. E-mail: alksarayah@uqam.ca



Agenda

25

- **Background to Functional Size Measurement (FSM) methods and their uses**
- **COSMIC FSM Method – ISO 19761**
- **Measurement Guidelines**
- ➔ ■ **‘The proof of the pudding is in the eating’: Good Estimation**
- **Automation of COSMIC measurement**
- **Conclusions**

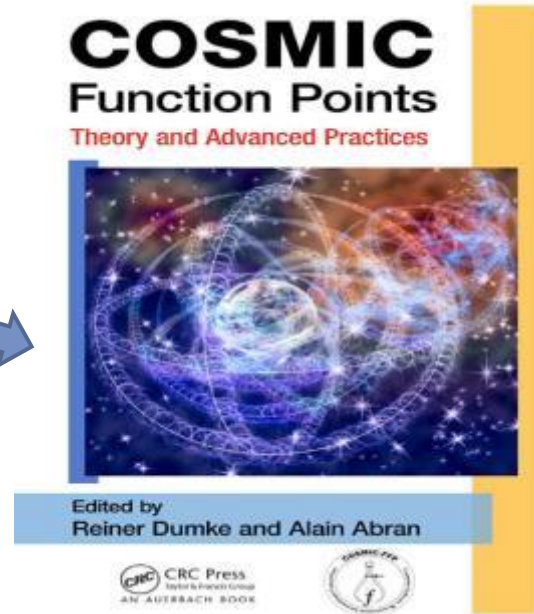
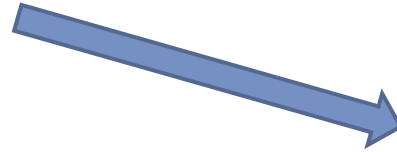
COSMIC data from Industry

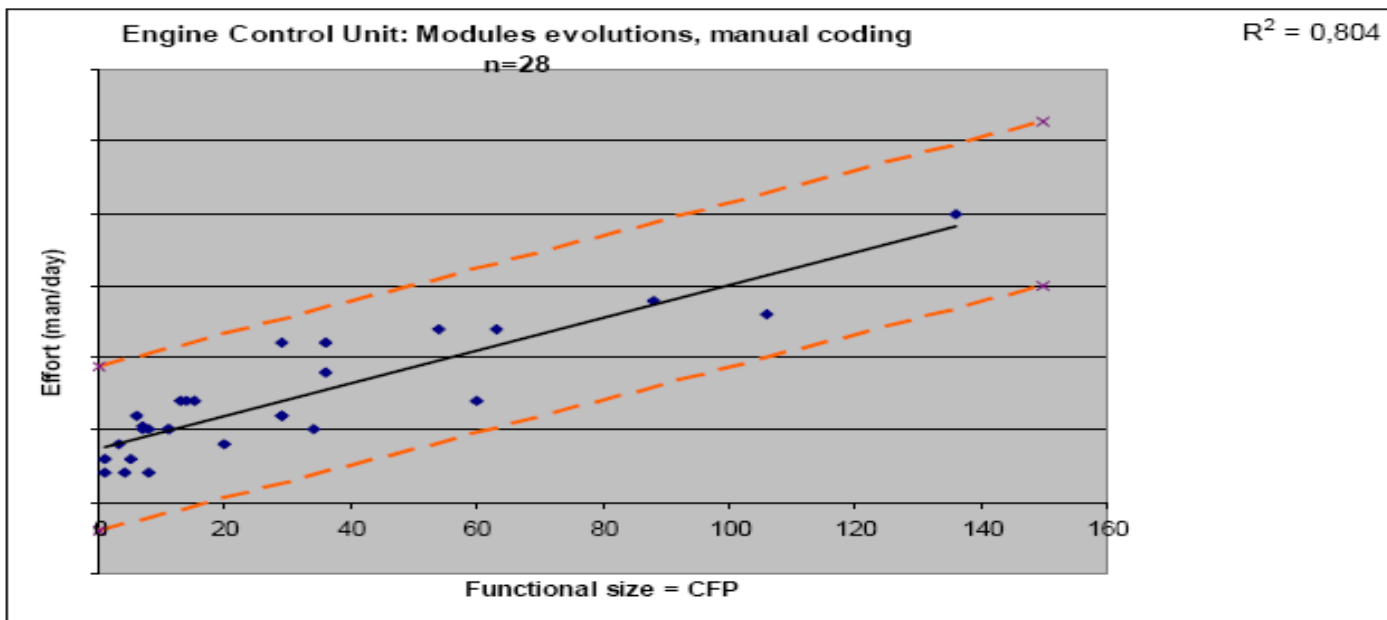
26

Practical experimentations with the
COSMIC method in Automotive
embedded software field

By: *Sophie Stern*

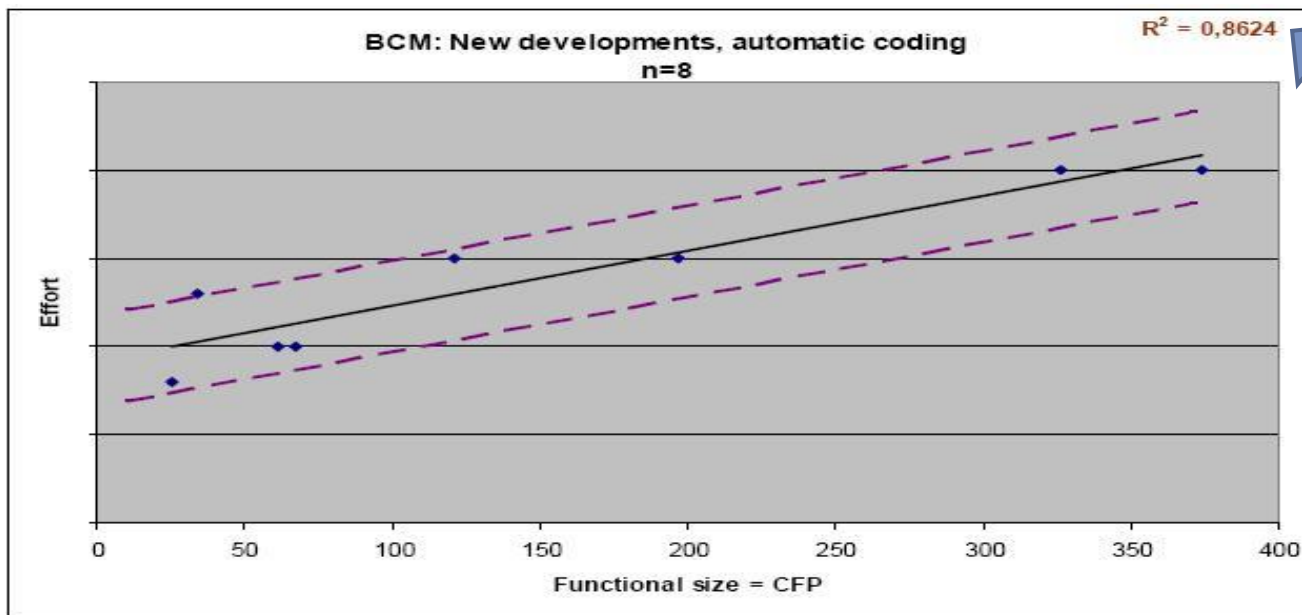
Renault





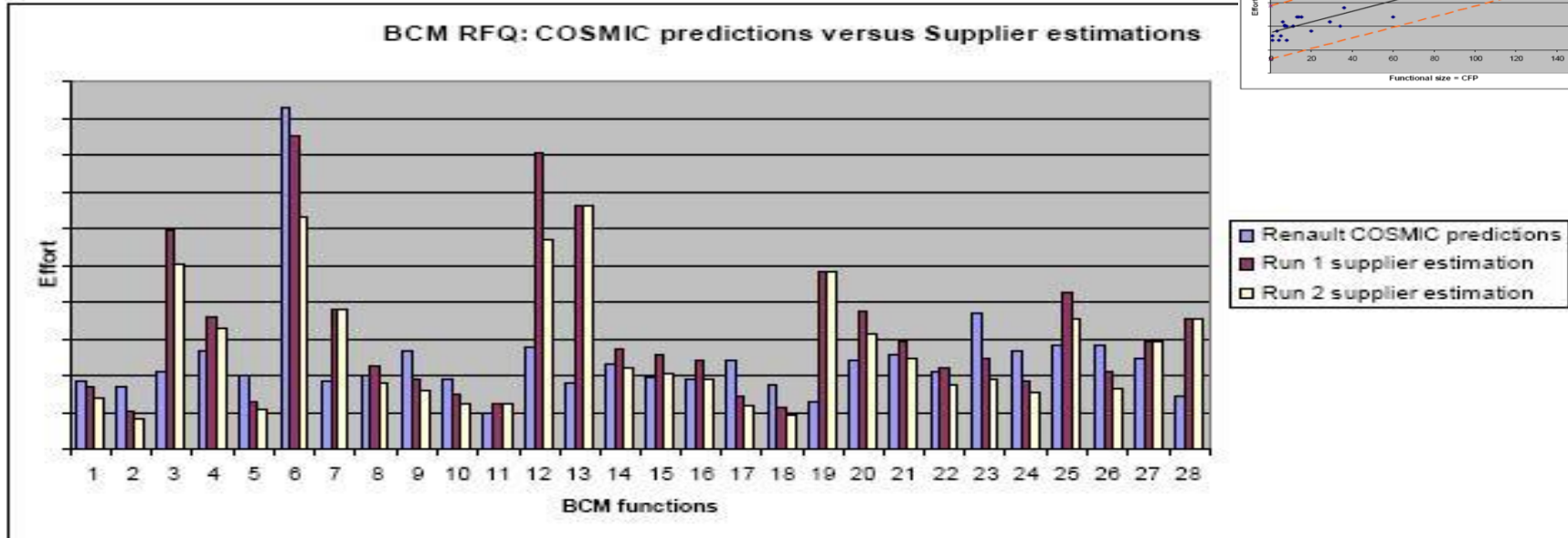
© Copyrights Renault 2012

Renault – 2012



© Copyrights Renault 2012

Renault: Estimation & Negotiations



© Copyrights Renault 2012

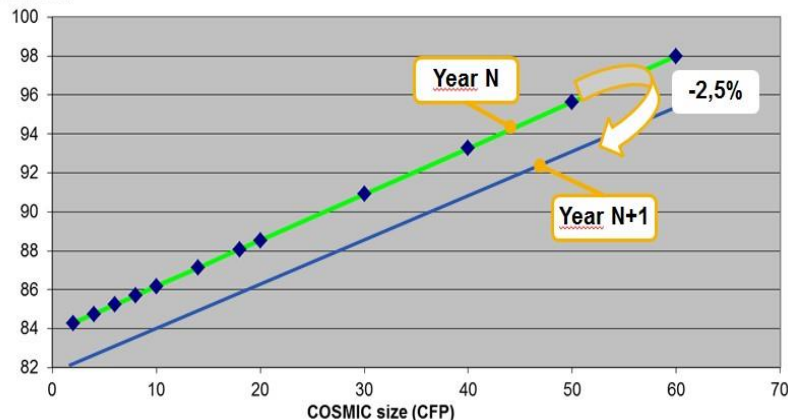


Renault - Remarkable cost estimation accuracy from its ECU software specifications

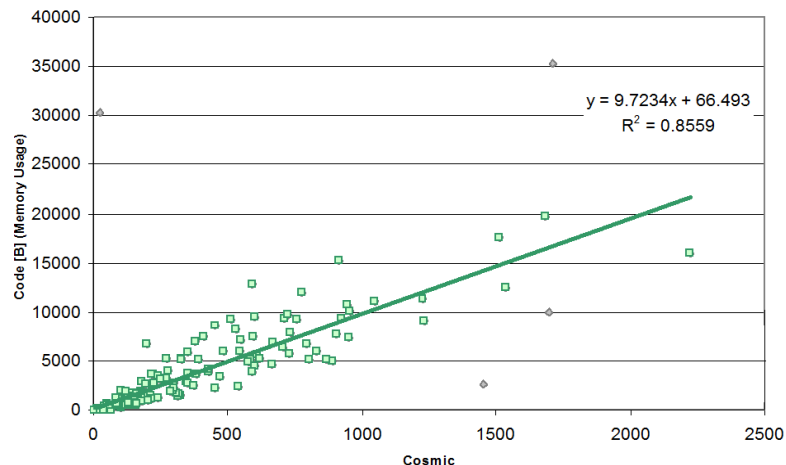
30

Workload (without unit) \Rightarrow K€

Purchase Department Negotiation



Cost vs size (CFP)

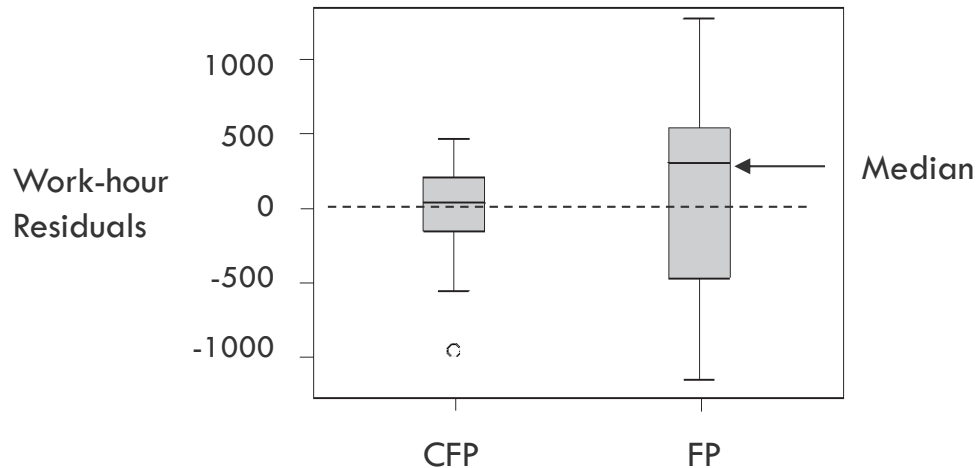


Memory size vs software size (CFP)



Case: Web effort estimation is more accurate with COSMIC than using classic FP

31



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
Sergio Di Martino^a, Filomena Ferrucci^{b,*}, Carmine Gravino^b, Federica Sarroc
[Information and Software Technology 72 \(2016\) 90–109](#)



Case: A Canadian supplier of security and surveillance software systems

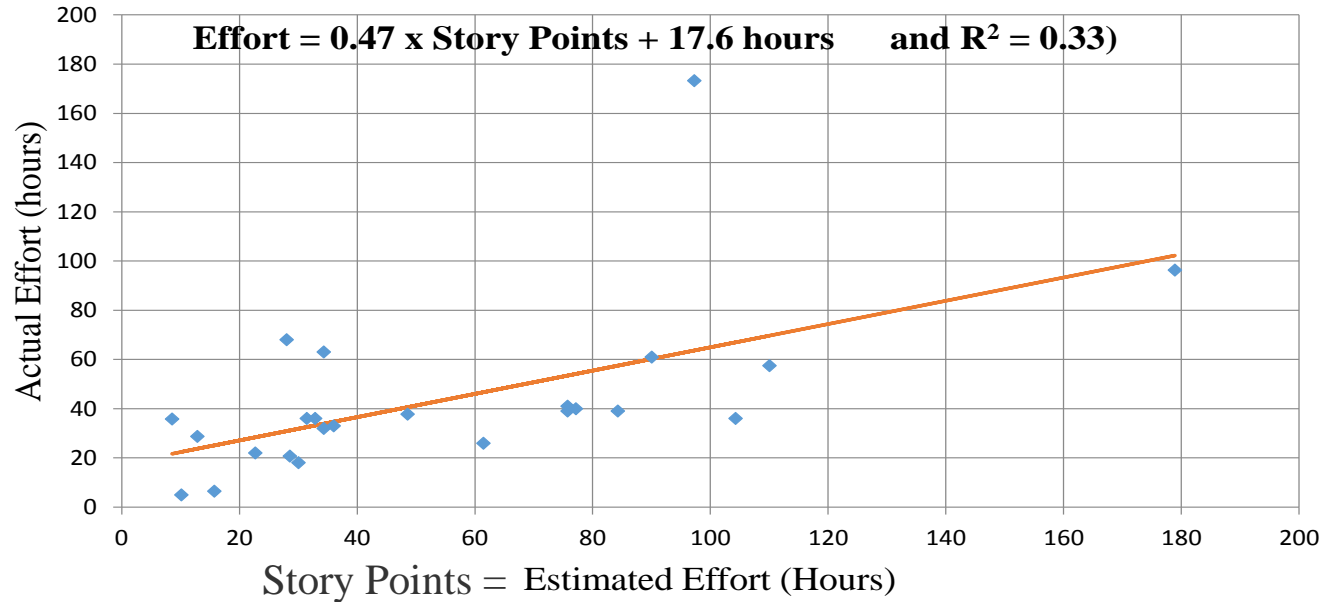
32

- Uses Scrum method
- Teams estimate tasks within each iteration in Story Points, and convert directly to effort in work-hours
- Study involved measurements on 24 tasks in nine iterations
 - Each task estimated in Story Points - Effort
 - 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

Effort vs Story Points (24 tasks) = a poor predictor of effort

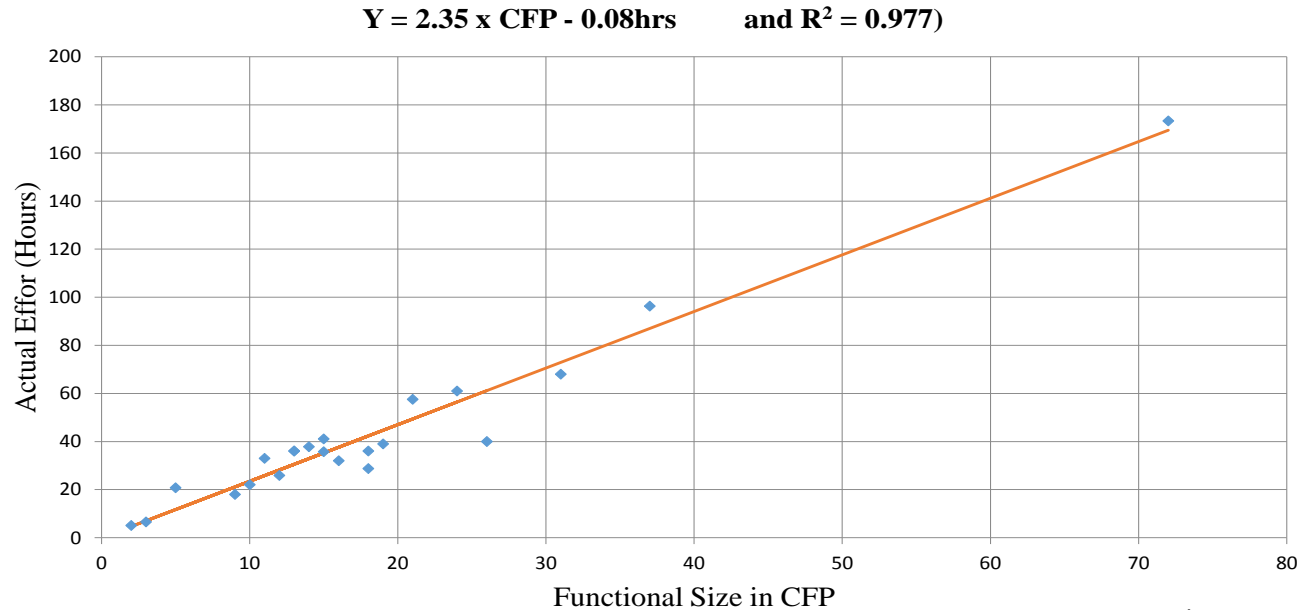
33





Effort vs COSMIC size is good for estimating

34



As a result of COSMIC measurement: two tasks were identified with very low effort/CFP.
These were found to involve significant software re-use, so were considered separately



A User view of 'COSMIC for Agile'

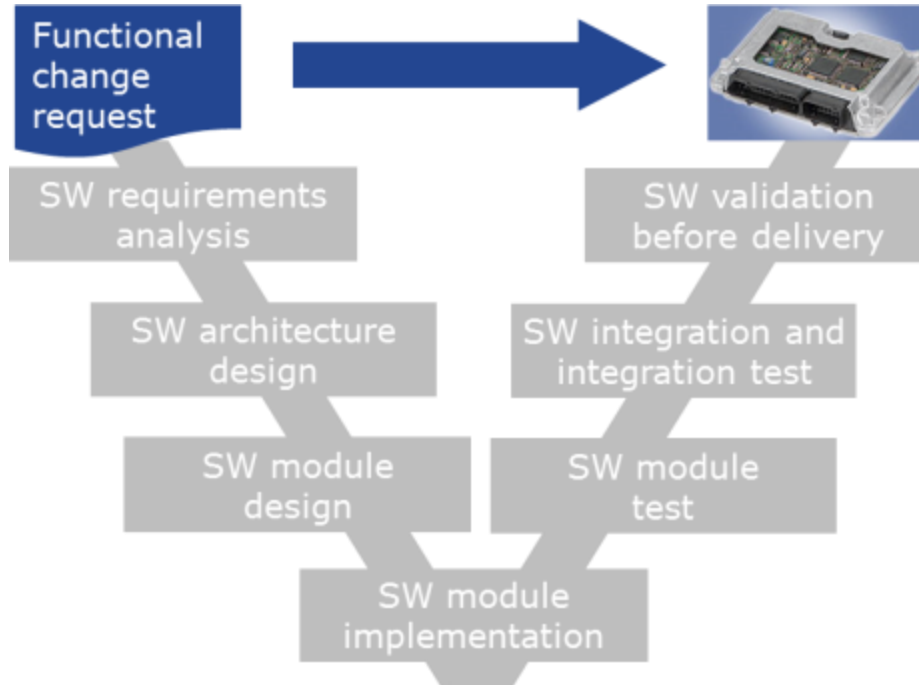
35

- *“We have found that adopting this approach provides us with excellent predictability and comparability across projects, teams, time and technologies.”*
- *The reality of achieving predictable project performance has driven me to investigate many methods of prediction. COSMIC is the method that lets me sleep at night.”*

Denis Krizanovic, Aon Australia, August 2014

Vector (Germany): Maintenance Constraints

36



Requirements and design specification:
Consistent level of documentation

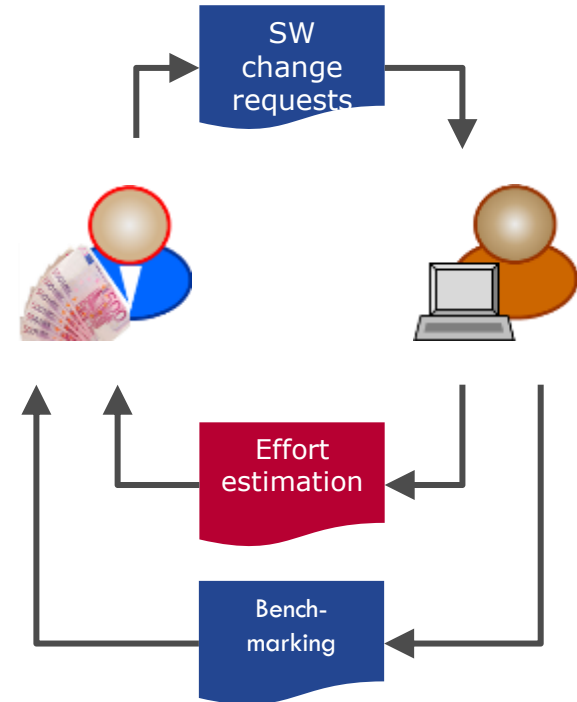
Modeling:
Same method,
notation, semantics
and visibility

Change documentation:
All changes are covered and clearly marked

Vector (Germany): COSMIC Benefits

37

- **Agreed model for measuring functional size**
- **Solid baseline for benchmarking**
- **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**





Vector (Germany): Results

38

- **Vector achieved with many clients a preciseness of 10-20% within one year of building the estimation program.**
- **Consider business impacts**
 - **Clearly distinguish goals, estimates & plans**
 - **Challenge results & improve your efficiency each year**
 - **Don't stay with same parameters for over 1 year.**
- **Establish repeatability**
 - **Immature processes invalidate your overall estimation & ruin trust**
 - **Establish a robust process to report & store data**
 - **Insufficient data quality & environmental constraints need experienced counting to avoid errors & weakening the method.**

Agenda

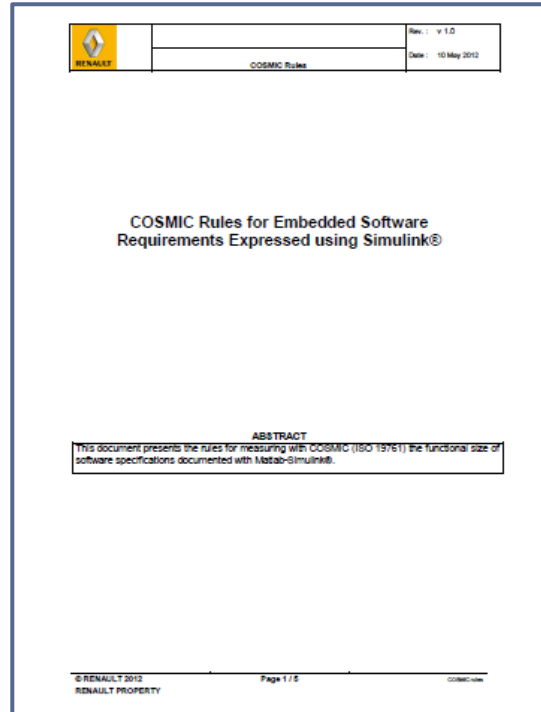
39

- **Background to Functional Size Measurement (FSM) methods and their uses**
- **COSMIC FSM Method – ISO 19761**
- **Measurement Guidelines**
- **‘The proof of the pudding is in the eating’: Good Estimation**
- ➔ ■ **Automation of COSMIC measurement**
- **Conclusions**



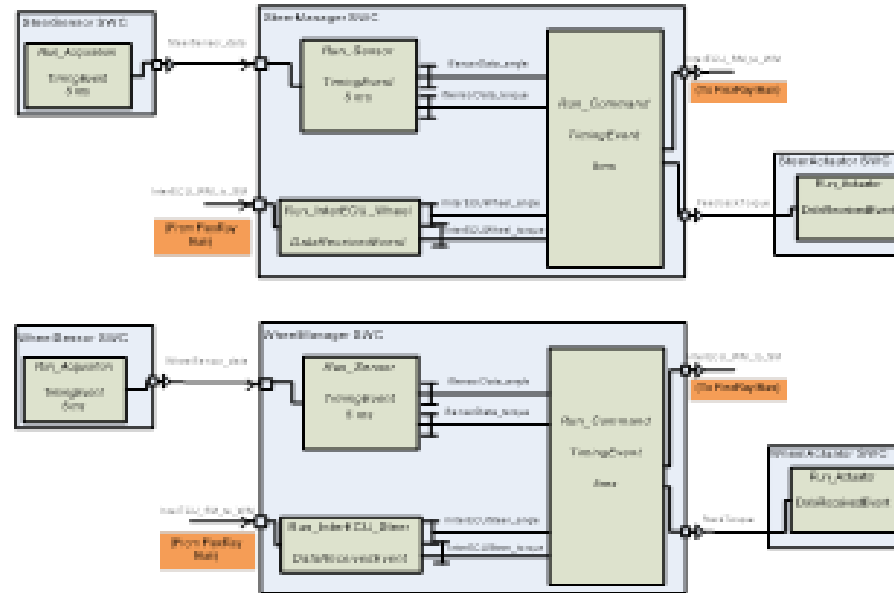
COSMIC specifications for Automation with Matlab-Simulink

40



Real-time embedded software specifications in graphical format – An example

41







Runnables inside the software components -Steer-by-Wire system

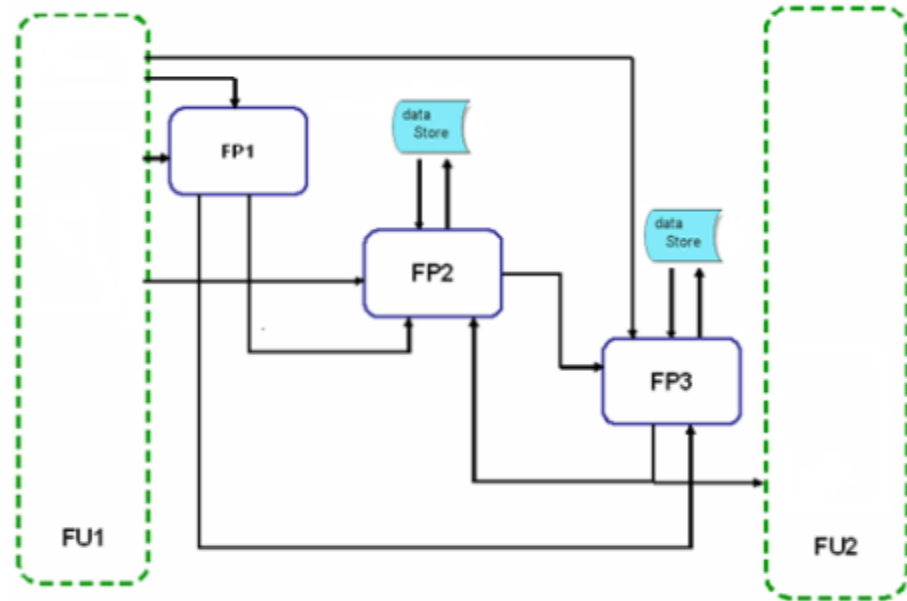
Ref. H. Soubra, and K. Chaaban, "Functional Size Measurement of Electronic Control Units Software Designed Following the AUTOSAR Standard: A Measurement Guideline Based on the COSMIC ISO 19761 Standard," IWSM-MENSURA Conference, Assisi (Italy), IEEE CS Press, 2012.

When Requirements are described with Graphs: Map to COSMIC

42

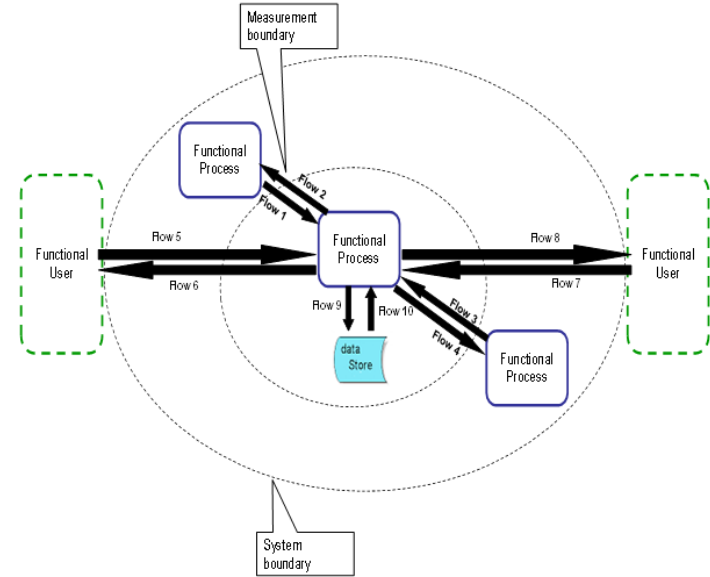
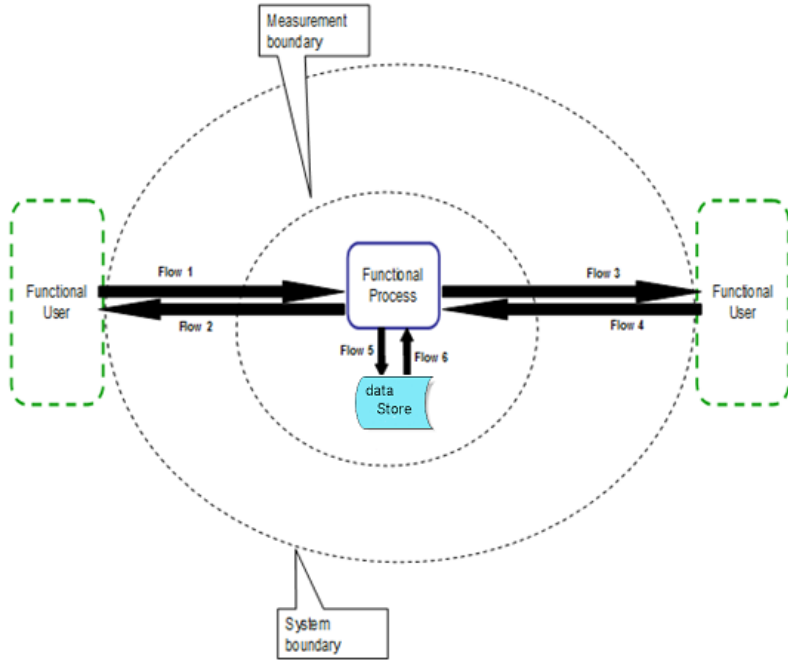
TABLE I.

COSMIC concepts	COSMIC abbreviation	Proposed graphical representation	Proposed graphical description
Functional user	<i>FU</i>		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



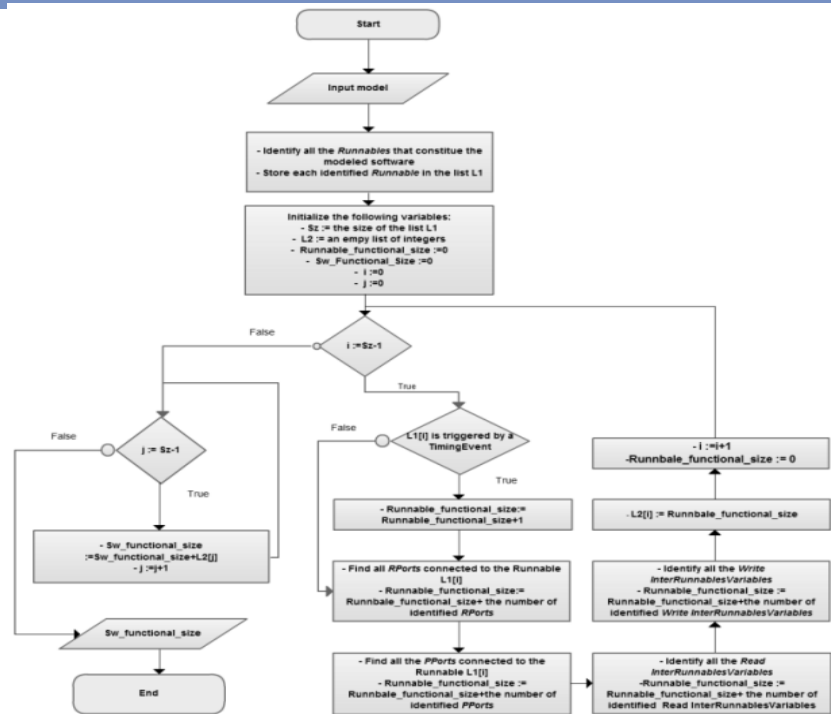
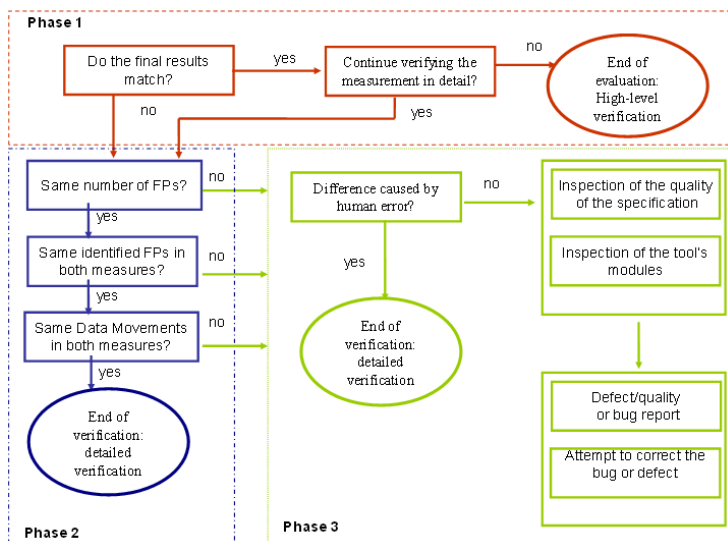
COSMIC representation of functional process (Single & Many) – For testing scenarios

43



3-Phase Verification Protocol of Automation Accuracy

44

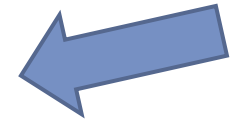


AUTOMATION ACCURACY REACHED WITH COSMIC

45

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_Acquisition	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

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 24th International Workshop on Software Measurement & 9th MENSURA Conference,
 Rotterdam (The Netherlands), Oct. 6-8, 2014, IEEE CS Press, pp. 23-31.



Agenda

46

- **Background to Functional Size Measurement (FSM) methods and their uses**
- **COSMIC FSM Method – ISO 19761**
- **Measurement Guidelines**
- **‘The proof of the pudding is in the eating’: Good Estimation**
- **Automation of COSMIC measurement**
- ➔ ■ **Conclusions**




The COSMIC method is very widely used

47

- **COSMIC Measurement Manual standard (11 languages)**
- **Size of user base is unknown**
 - **Of known users, 50% are software houses**
 - **Adopted by two Governments (Mexico, Poland)**
 - **> 30,000 downloads of research & conference papers**
- **+ 600 certification exam holders (notably Brazil, China, India, Italy, Poland, Turkey)**
- **Two active forums (on LinkedIn CUG, www.cosmic-sizing.org)**

Summary of benefits

48

- **Free, open**
- **Fundamental SE Principles:  future-proof, stable**
- **Very wide applicability**
- **Proven value for performance measurement & estimating**
- **ISO standard & GAO ¹, NIST ² documents**
- **Can be automated with very high accuracy & traceability**

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

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



Acknowledgements

50

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

There is a well-defined Measurement Process

51

