Presented at the 2016 International Training Symposium: www.iceaaonline.com/bristol2016

COSMIC

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)

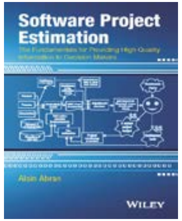
Presenter background: Alain Abran

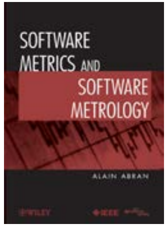
20 years

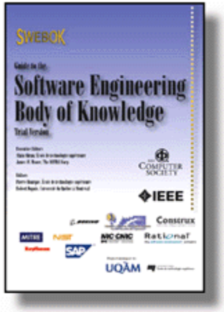
- Development
- Maintenance
- Process Improvement

20 years

+ 35 PhD







ISO: 19761, 9216, 25000, 15939, 14143, 19759

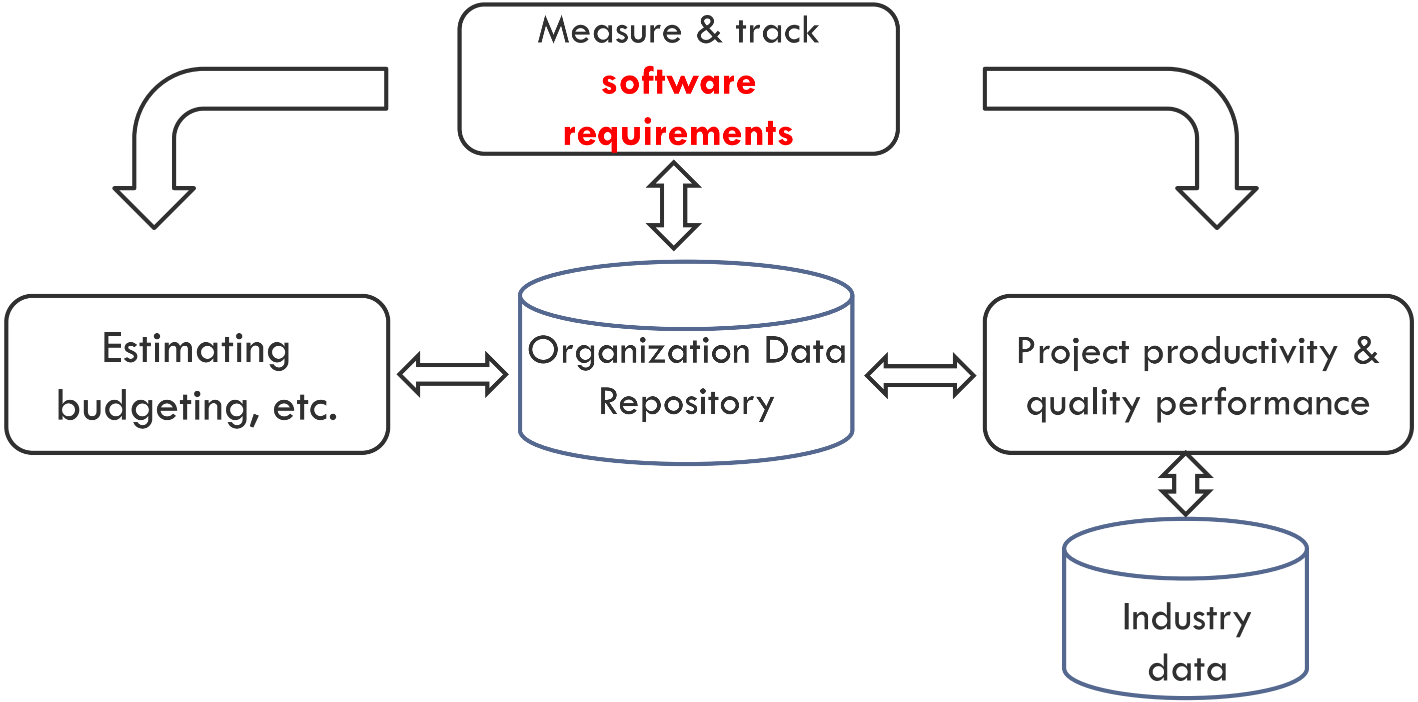
COSMIC

Agenda

- 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

COSMIC

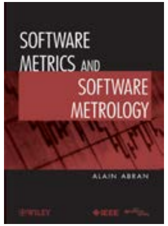
Objective: we want to use performance data for estimating future projects



Software Sizing Options

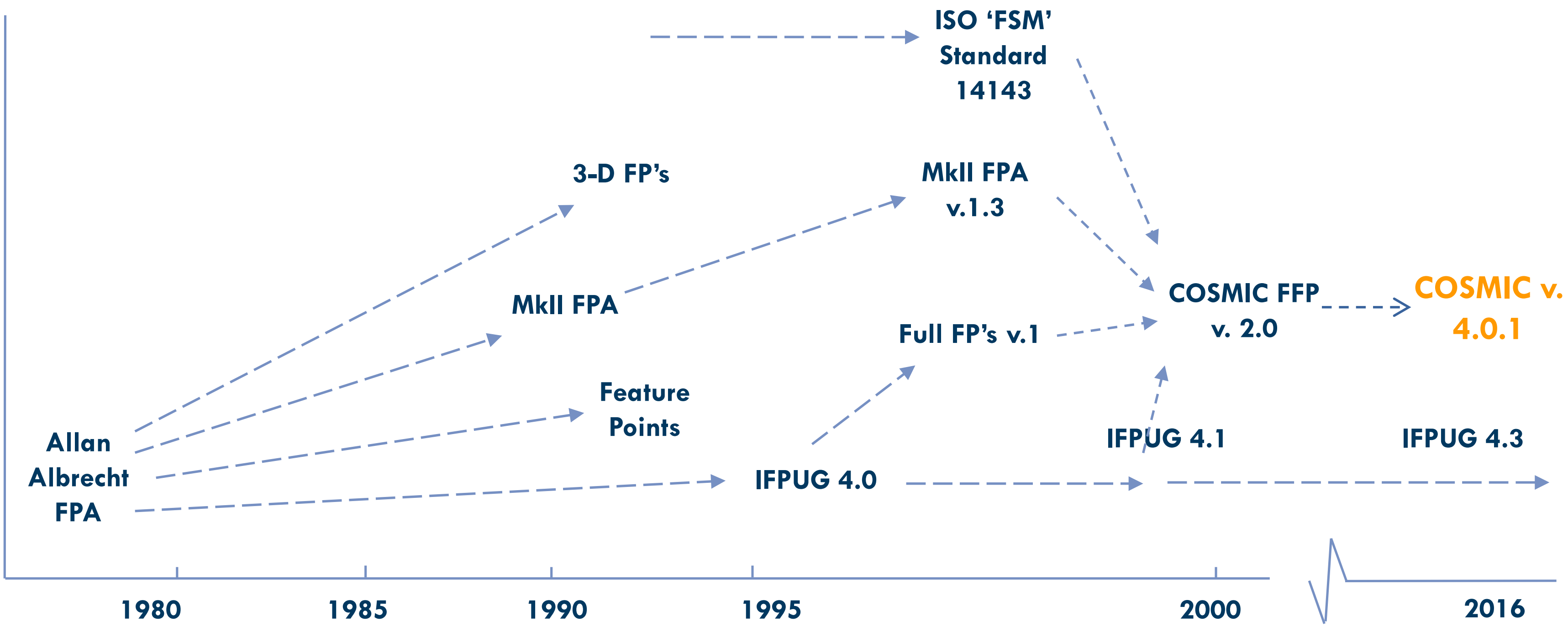
Sizing method options:

- 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 Mathematically invalid?
- Story Points (Planning Poker):
  - X Entirely Subjective & Benchmarking impossible:
    - **unaccountability**
- Functional size
  - ✓ (Function Points):
  - ✓ International standard methods
  - ✓ Technology-independent

Function Points have been around for a long time: + 35 years!



COSMIC

Agenda

- 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

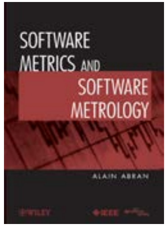
COSMIC

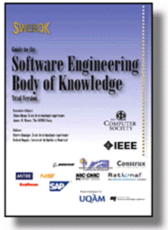
Key Concepts of Measurements

- Measurement of what is common
  - not what is different: measurement of a single concept independent of contexts
- Measurement standards – for all people
- Standard Unit of measurement ('étalon') to ensure consistency-repeatability-reproducibility, etc.
- Measuring instruments:
  - Manual – procedures & guidelines
  - Partially or Totally automated with measuring instruments

COSMIC Method

- 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)





COSMIC

Software is an 'intellectual' product: can we measure it?

- Time: can we 'see' or 'touch' time?
  - Answer = …………
- Distance: is there a unique distance between 2 cities?
  - Answer = ………….
- Do we need to 'see' something to measure it?
  - Answer = …………
- Can we measure something before it exists?
  - Answer = ………….

COSMIC

Software is an ‘intellectual’ product: can we measure it?

- Time: can we ‘see’ or ‘touch’ time?
  - Answer = No (.. but measuring instruments have been built…)
- Distance: is there a unique distance between 2 cities?
  - Answer: It depends (by road, by car, by train, by highways, by plane)
- Do we need to ‘see’ something to measure it?
  - Answer = No (…ex. microscope)
- Can we measure something before it exists?
  - Answer = Yes (….from their representation in models & plans)

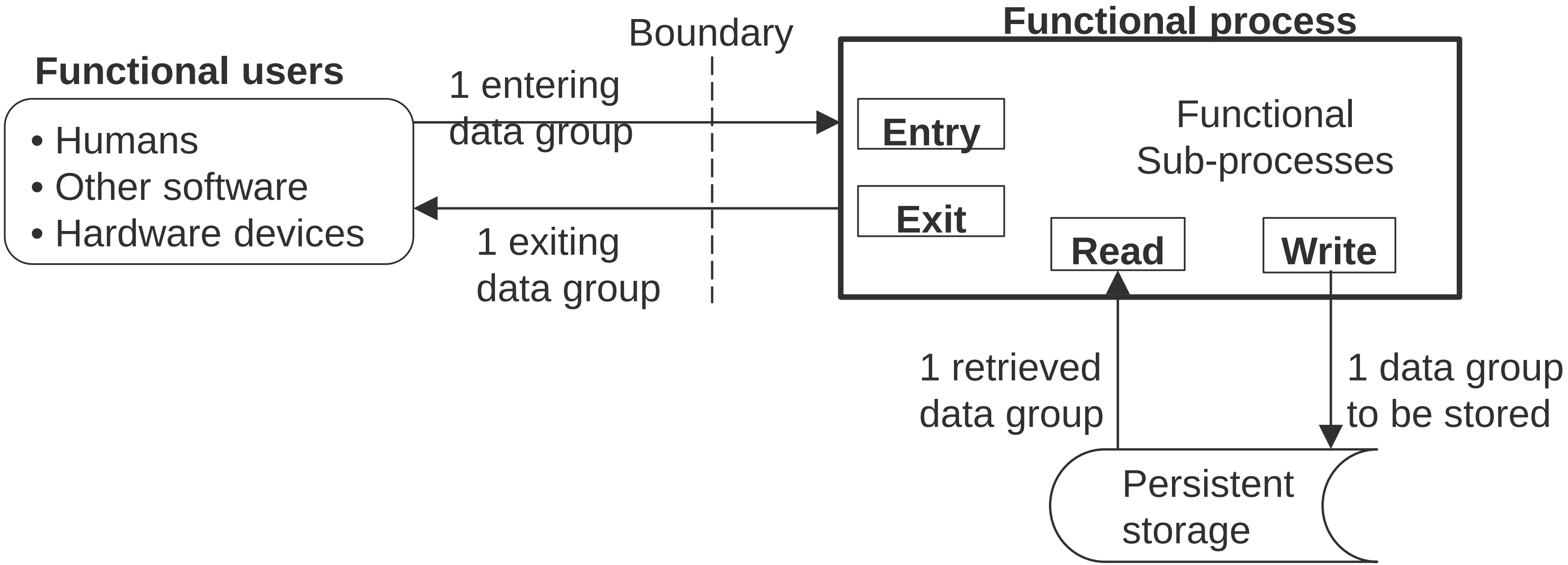
COSMIC

COSMIC view of software

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

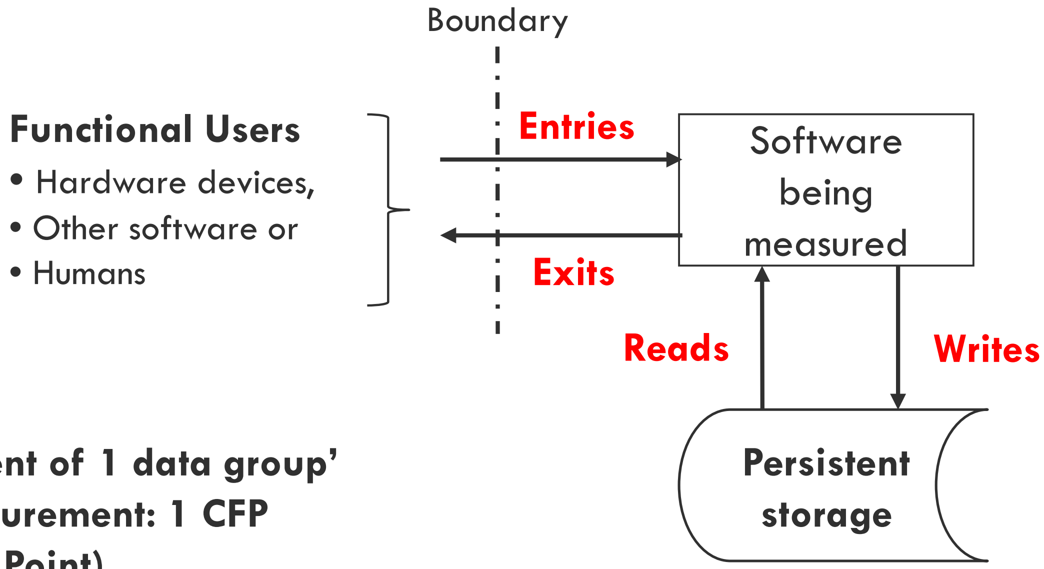
COSMIC

All software does this!



COSMIC

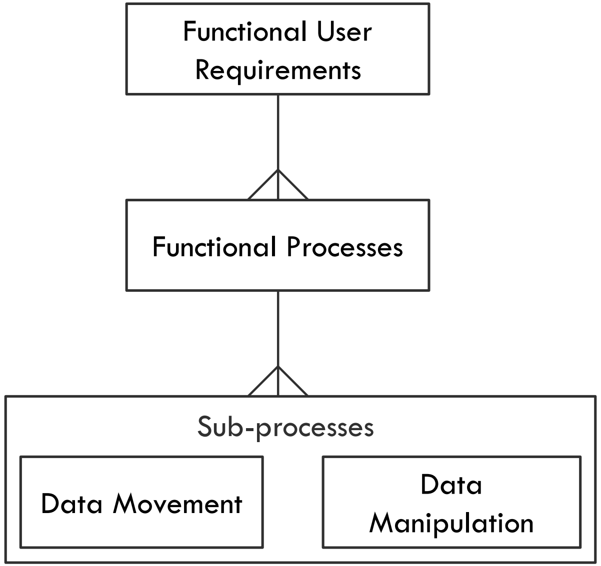
4 types of 'Data Movement'



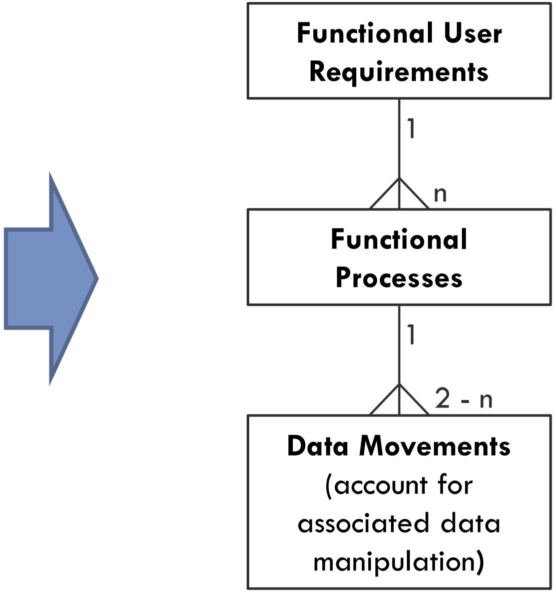
The 'Data Movement of 1 data group' is the unit of measurement: 1 CFP (COSMIC Function Point)

All software Functional User Requirements can be broken down into functional processes

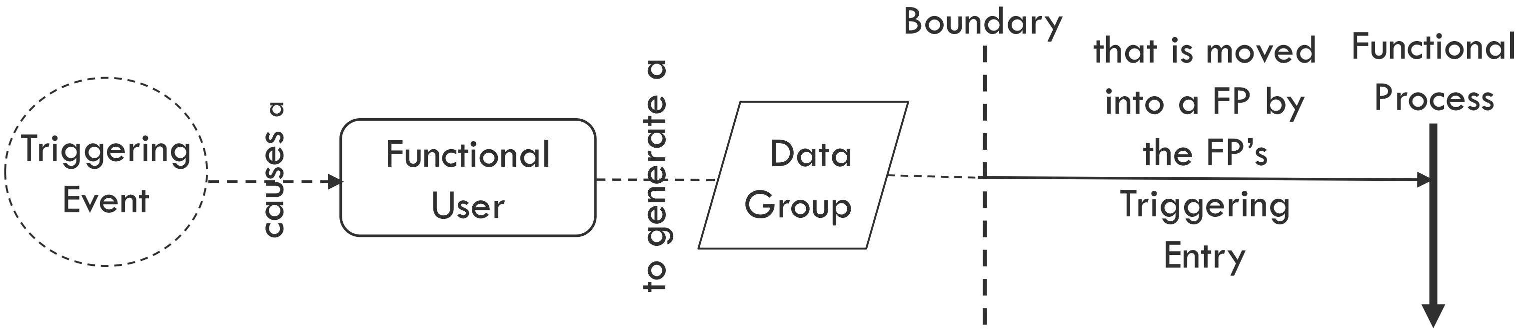
Theory:



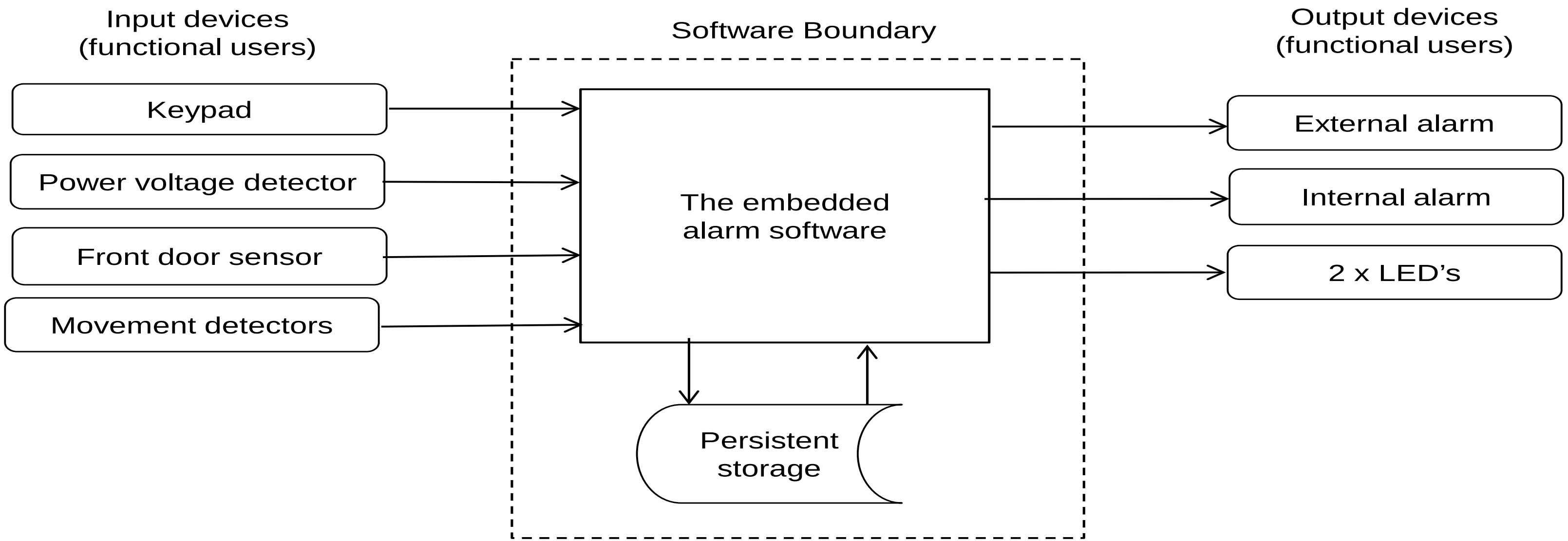
In practice:



A Functional Process responds to an 'Event' that a 'Functional User' detects or generates



Example: Intruder Alarm System



COSMIC

Intruder Alarm System

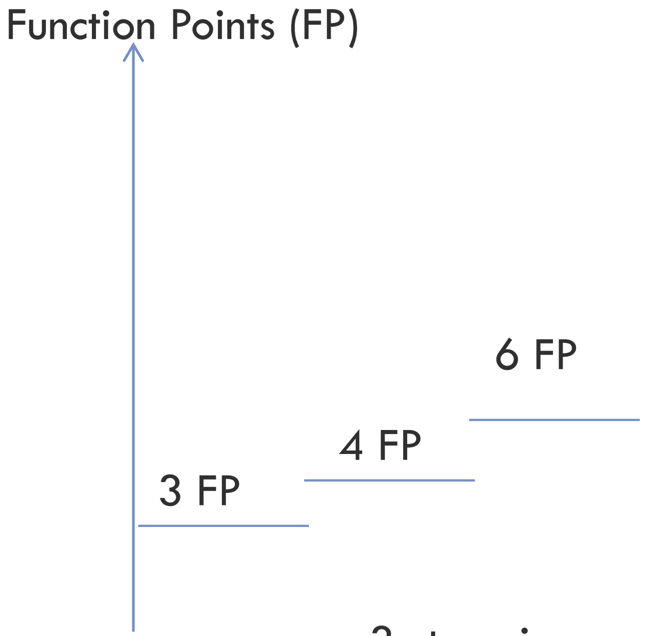
Functional process: Possible intruder detected.

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

1st Generation of Function Points: Step Functions!



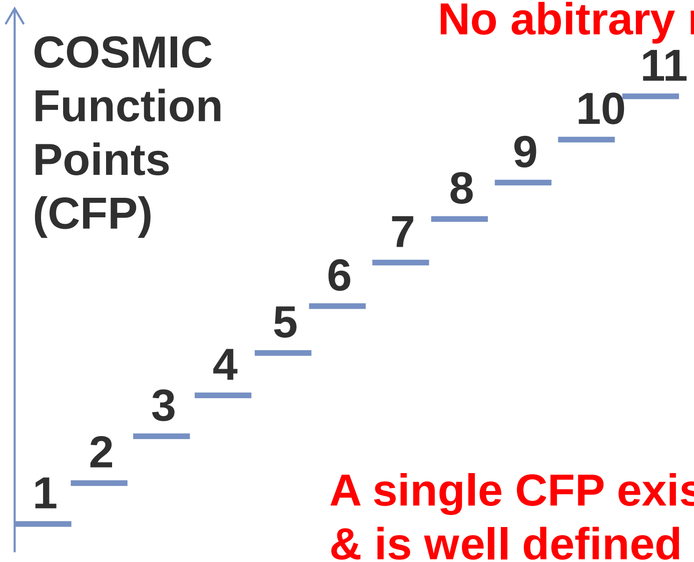
Key limitations:

- **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

No abitrary max



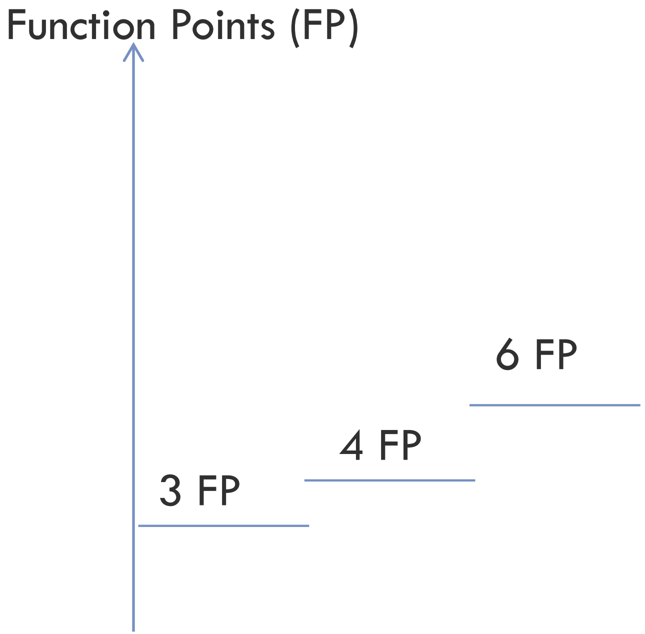
A single CFP exists
& is well defined

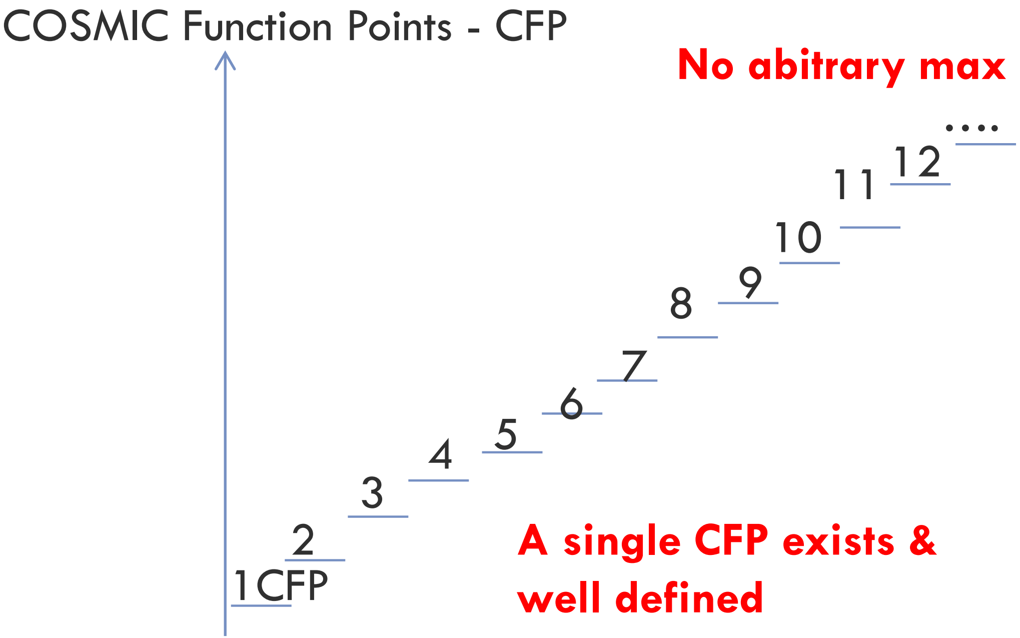
COSMIC sizes are measured on a true ratio scale

- 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

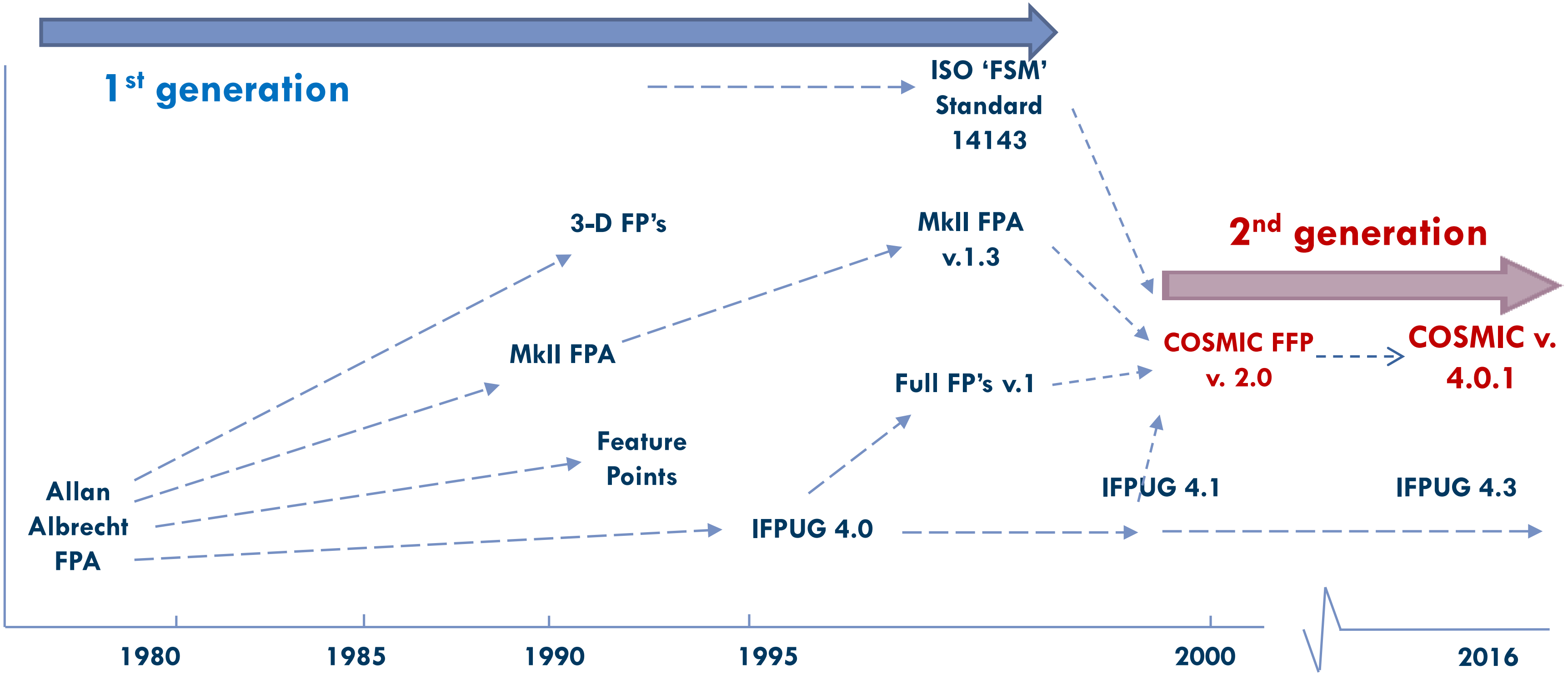
- Open, freely available (via www.cosmic-sizing.org)

1st and 2nd Generations of FSM

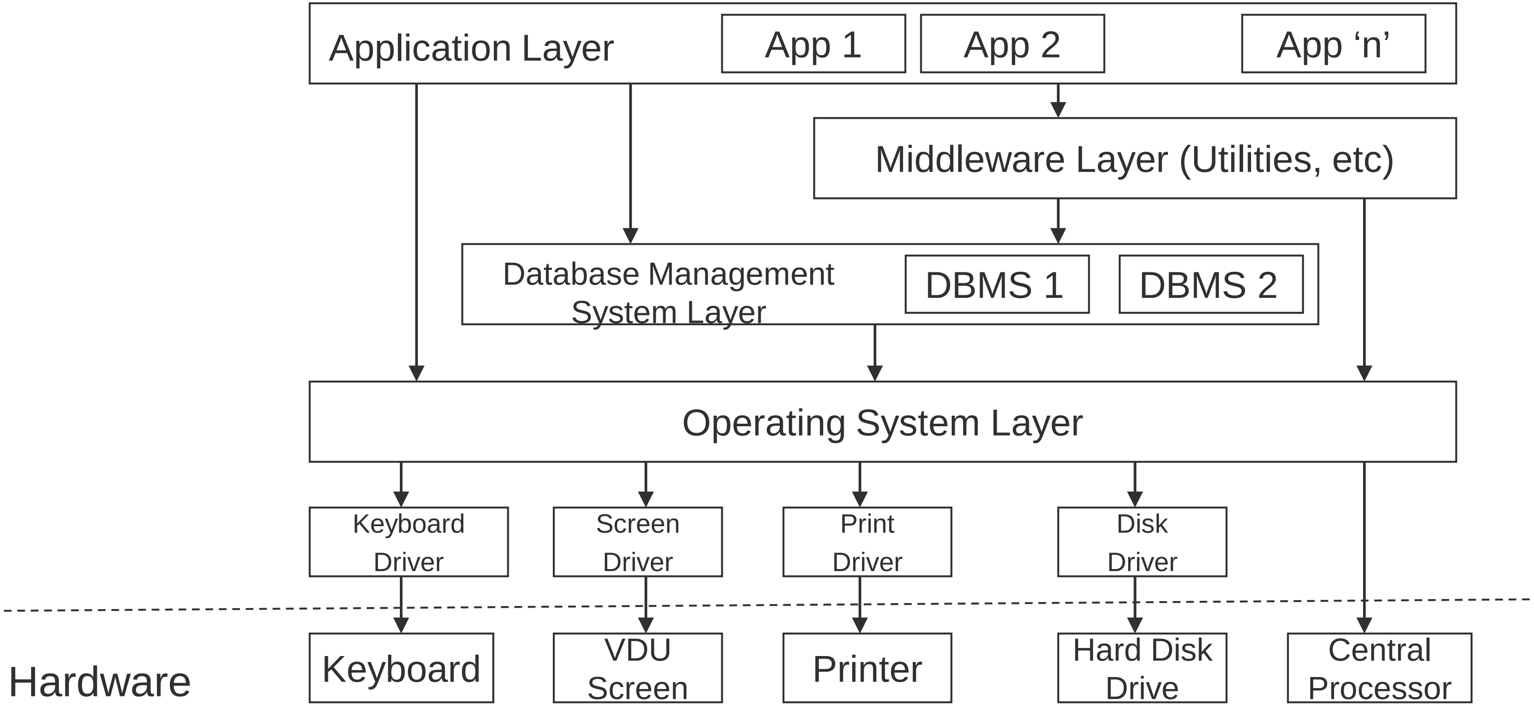




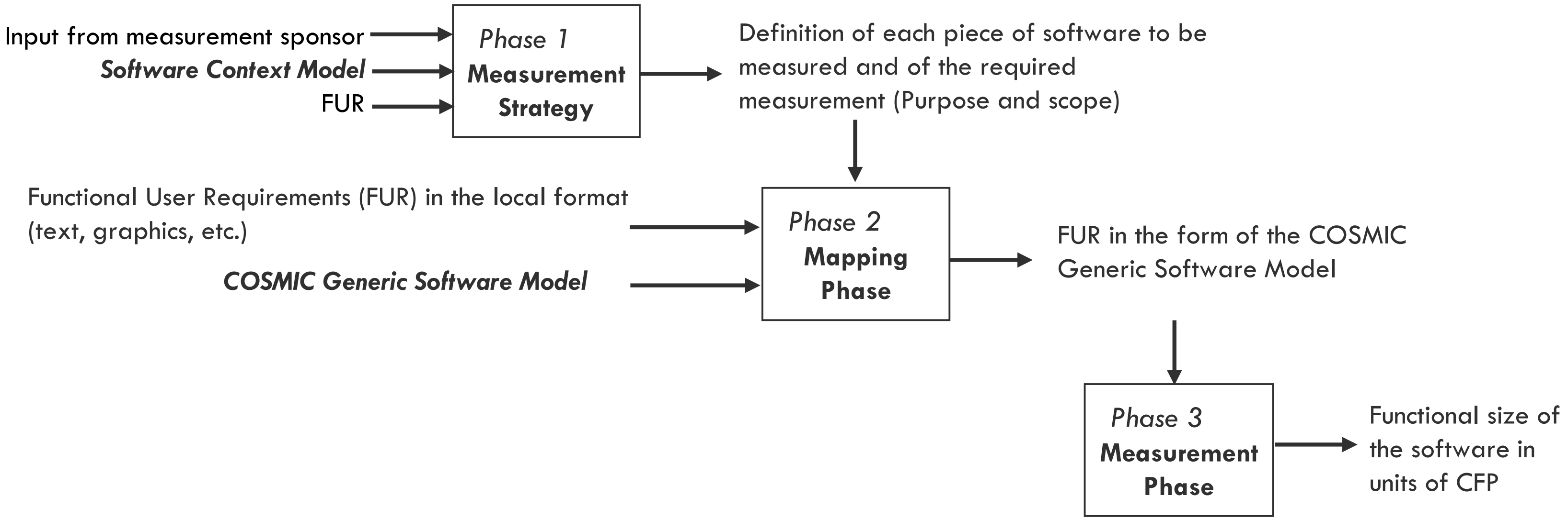
1st & 2nd generation of Function Points Methods



COSMIC - at any level of software requirements



There is a well-defined Measurement Process



COSMIC

Agenda

- 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

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

The COSMIC Functional Size Measurement Method
Version 3.0.1

Guideline for assuring the accuracy of measurements

VERSION 0.93
February 2011

Guidelines by Application Domains

- Business applications
- Real-time software
- Data Warehouse software
- SOA software
- Mobile apps

and for Agile Developments

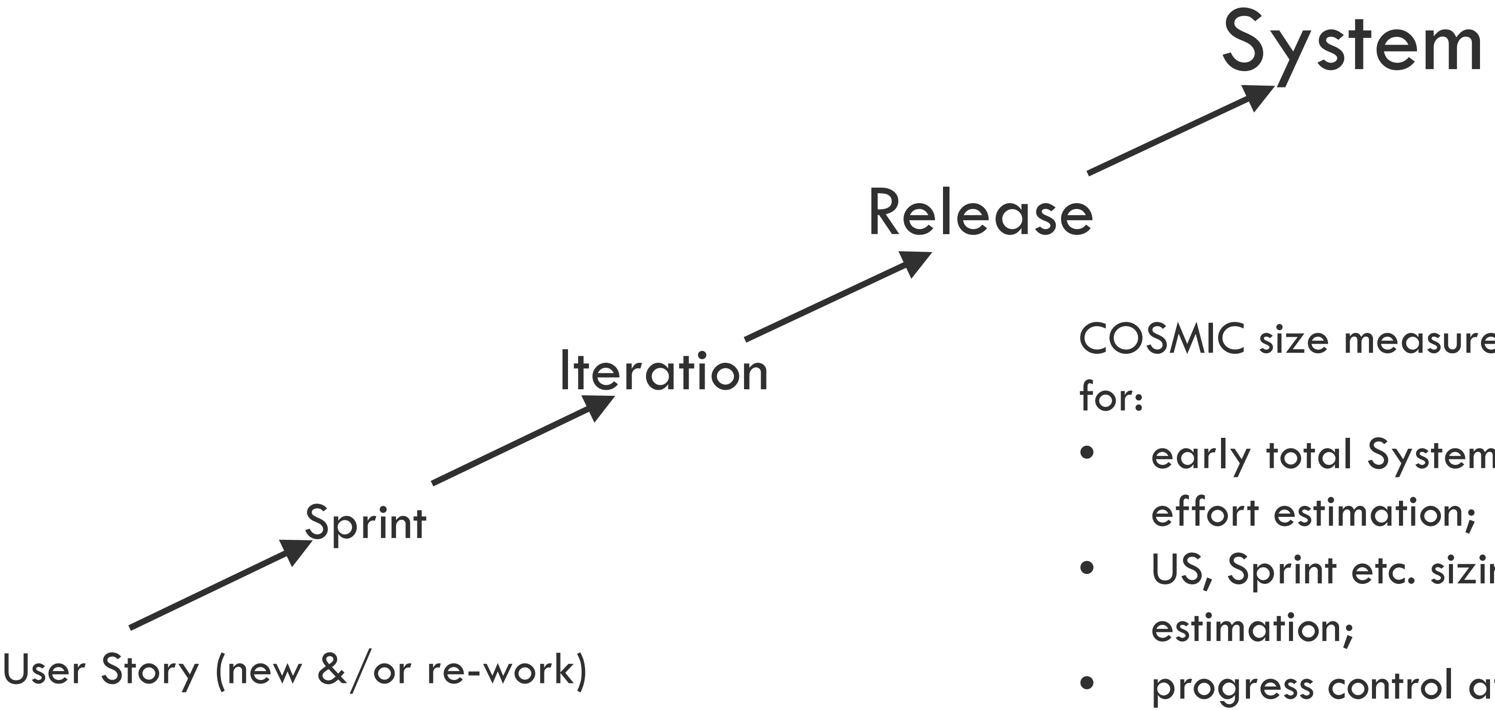

COSMIC

The COSMIC Functional Size Measurement Method
Version 4.0.1

Guideline for Sizing
Business Application Software

VERSION 1.3a
Febuary 2016

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



COSMIC size measurement is usable for:

- early total System sizing and effort estimation;
- US, Sprint etc. sizing and estimation;
- progress control at any level.

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

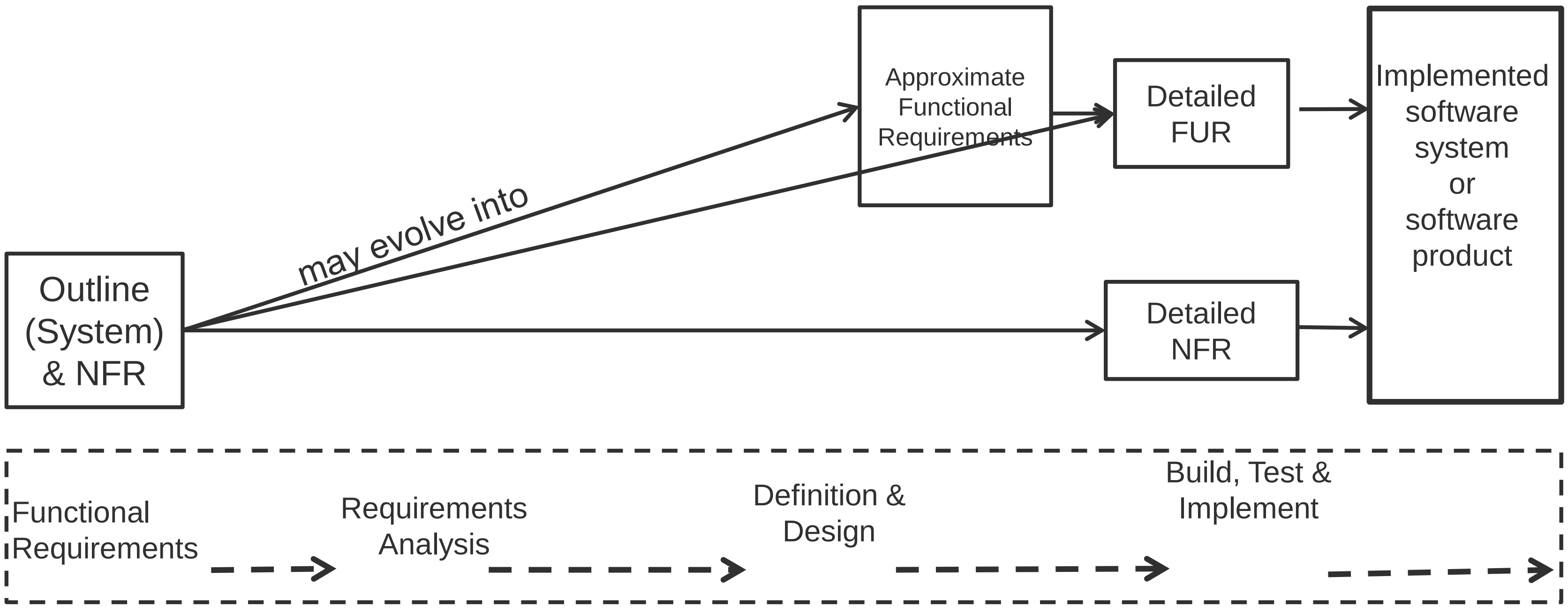
Guideline on Non-Functional & Project Requirements

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

Version 1.

November 2015

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.

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

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]

[a] Software Engineering and Information Technology Department, École de technologie supérieure (ETS), University of Quebec, Canada
[b] Software Engineering Department, Hashemite University (HU), Jordan
[c] Computer Science Department, University of Alcalá, Spain

ARTICLE INFO

Article history:
Received 4 April 2012
Received in revised form 4 October 2012
Accepted 14 November 2012
Available online 4 January 2013

Keywords:
Software Engineering Requirements
Non-functional requirements
Portability requirements
COSMIC-ISO 19761
ECSS Standards

ABSTRACT

In the system requirements phase, the non-functional requirements (NFR) are often captured only generically at a fairly high level, and they do not yet include the levels 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 portability requirements as one of sixteen types of non functional requirements (NFR) for embedded and real-time software. A number of portability-related concepts are dispersed throughout the ECSS, IEEE-830, ISO 9126, ISO 24765, and ISO 2382-1 standards to describe, at varying levels of detail, the various types of candidate portability requirements at the system, software, and hardware levels. This paper organizes these dispersed portability concepts and terms into a standards-based reference framework of system portability requirements. The availability of this framework can facilitate the early identification and specification of the system portability NFR and their detailed allocation as specific portability functions to be handled by the specified allocation to hardware or software, or a specific combination of the two. The approach selected in this research for the structure of this reference framework is based on the generic model of software proposed in the COSMIC-ISO 19761 model, thereby allowing the functional size of the portability requirements allocated to software to be measured.

1. Introduction

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 system that satisfies the stakeholder expectations and needs.

II. RELATED WORK

ISBN: 978-1-61804-290-3

COSMIC

Agenda

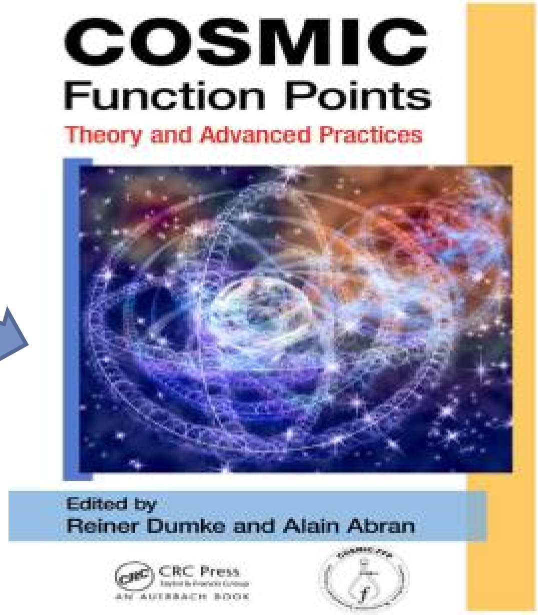
- Background to Functional Size Measurement (FSM) methods and their uses
- COSMIC FSM Method – ISO 19761
- Measurement Guidelines
- ‘The proof of the pudding is in the eating’: Good Estimation
- Automation of COSMIC measurement
- Conclusions

COSMIC data from Industry

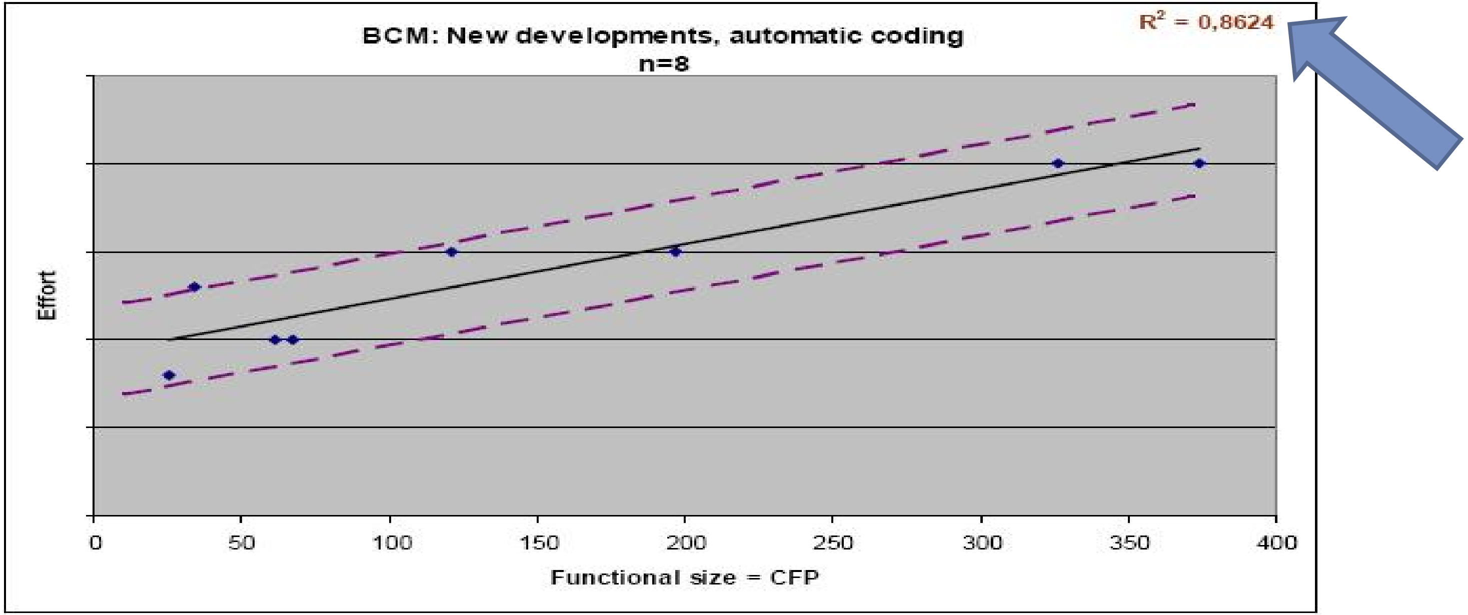
Practical experimentations with the COSMIC method in Automotive embedded software field

By: Sophie Stern

Renault

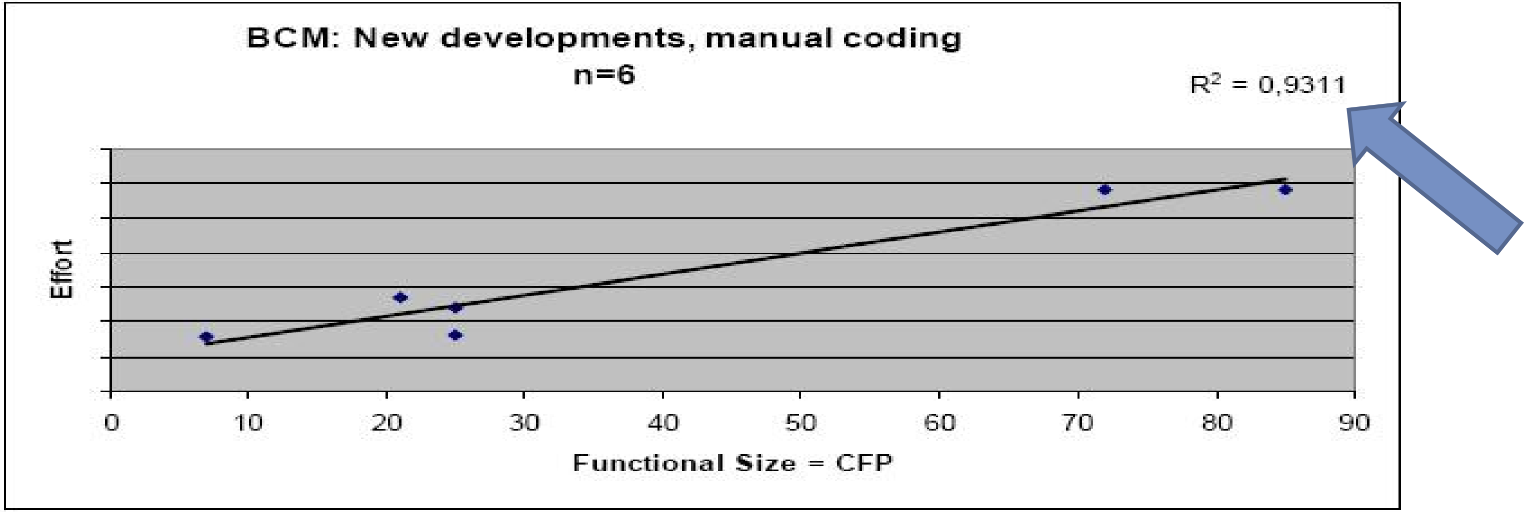


Renault – 2012

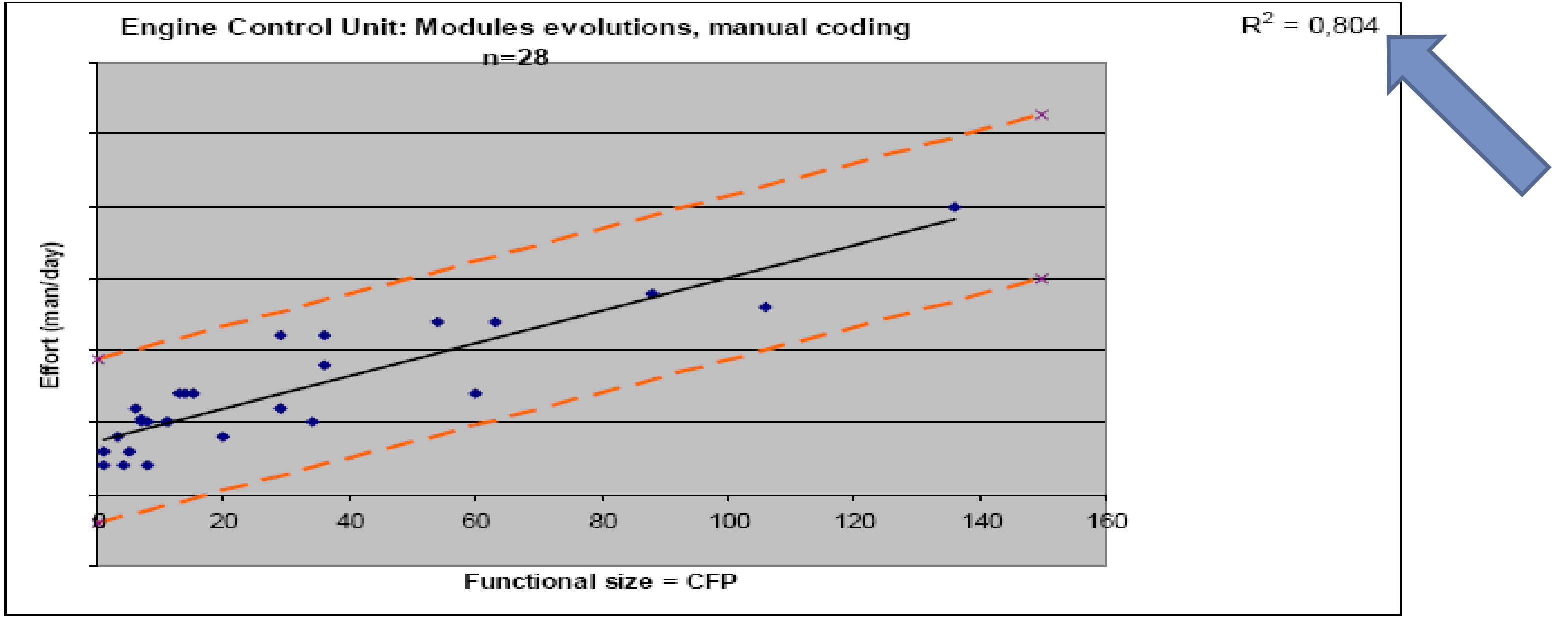


COSMIC

Renault – 2012

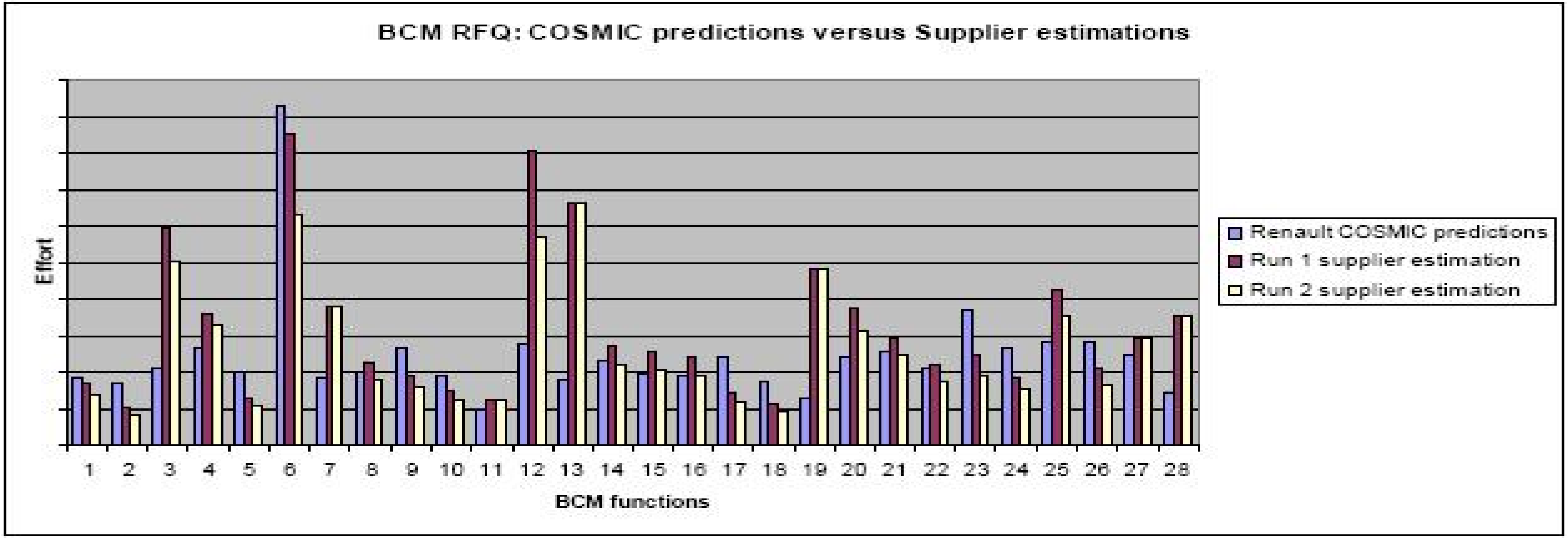


COSMIC

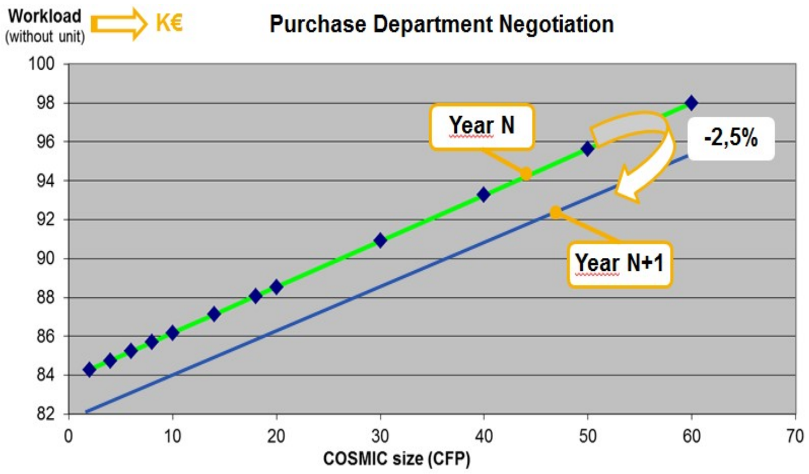

Engine Control Unit: Modules evolutions, manual coding
n=28
R^2 = 0,804
Effort (man/day)
0
20
40
60
80
100
120
140
160
Functional size = CFP

COSMIC

Renault: Estimation & Negociations

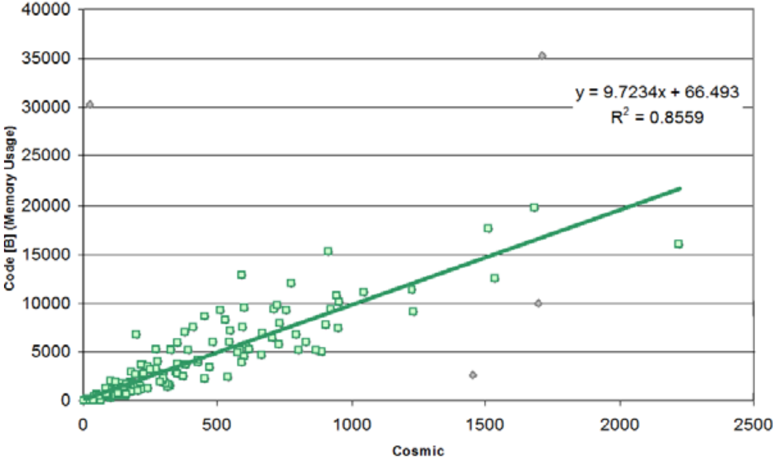


Renault - Remarkable cost estimation accuracy from its ECU software specifications

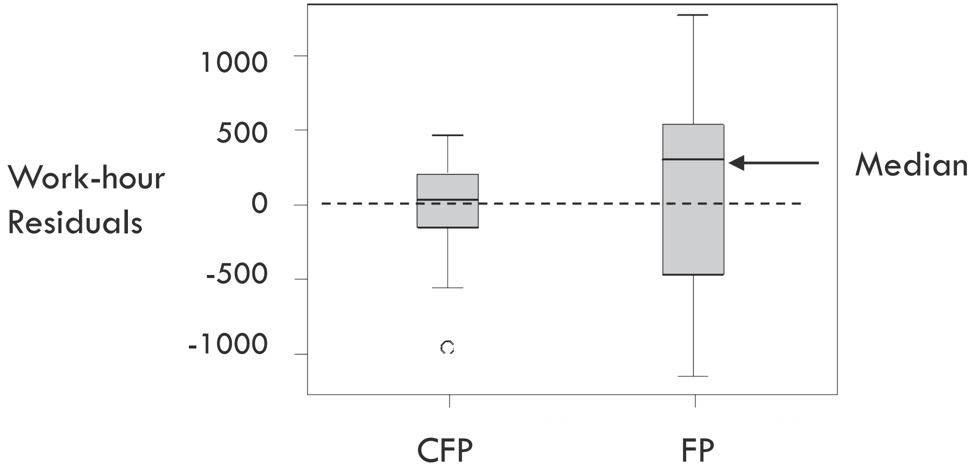


Cost vs size (CFP)

Memory size vs software size (CFP)



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



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 Martinoa, Filomena Ferruccib,*, Carmine Gravinob, Federica Sarroc
Information and Software Technology 72 (2016) 90–109

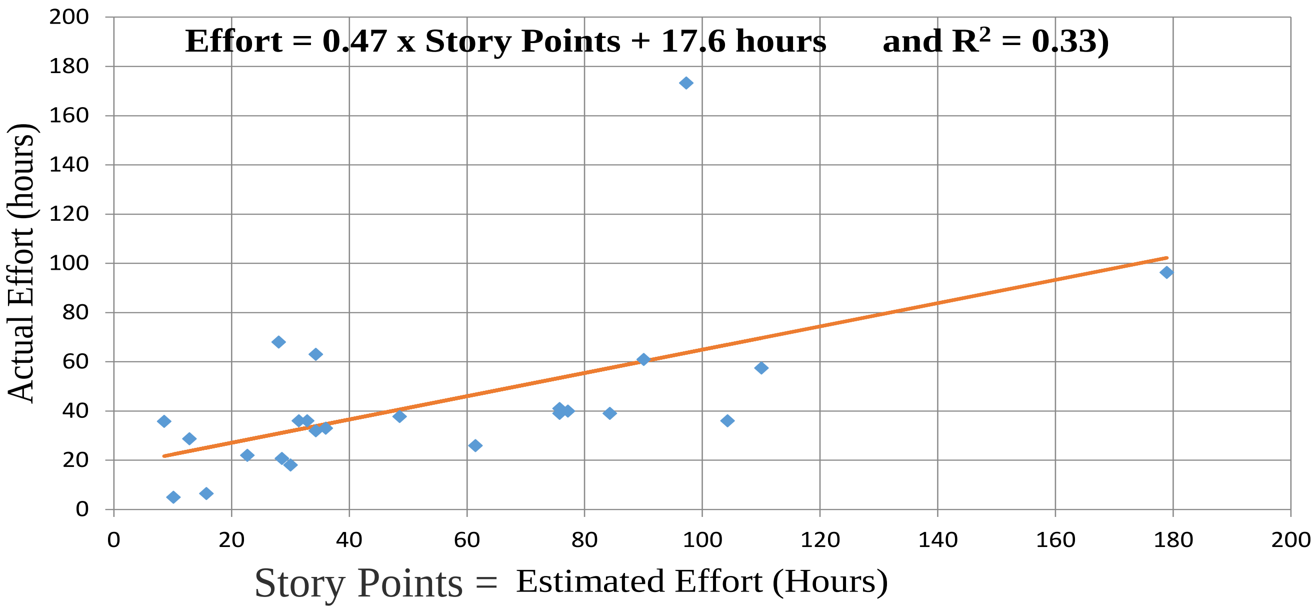
Case: A Canadian supplier of security and surveillance software systems

- 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

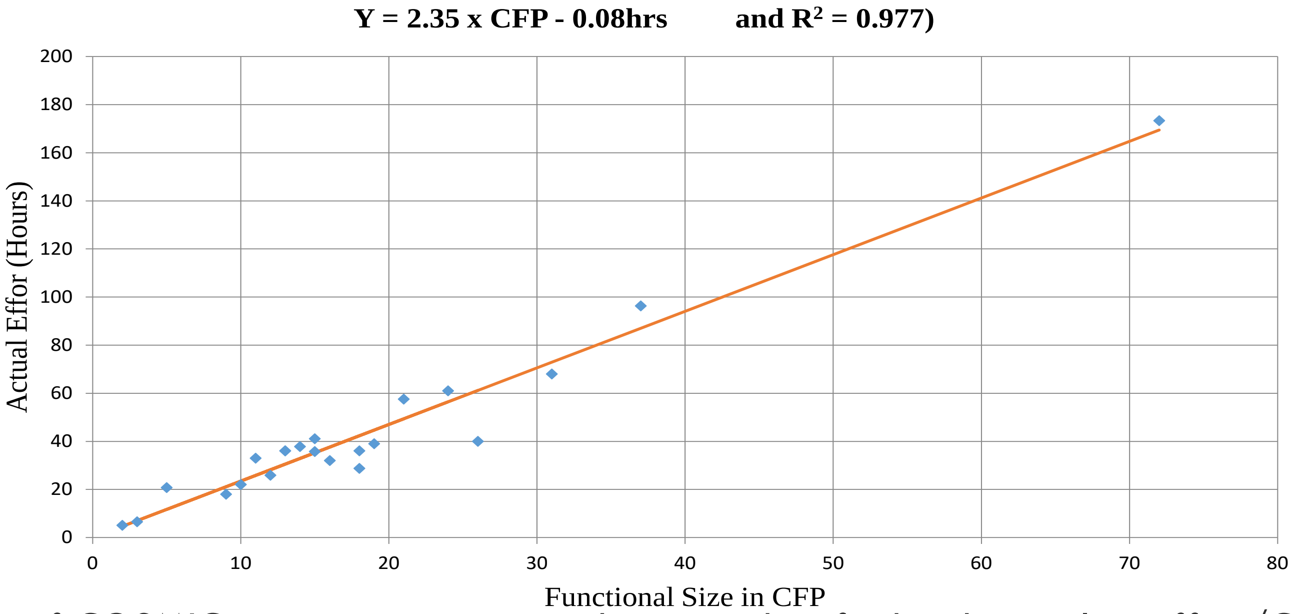
Ref. 'Effort Estimation with Story Points and COSMIC Function Points - An Industry Case Study',
C. Commeyne, A. Abran, R. Djouab. Obtainable from www.cosmic-sizing.org 'Software Measurement News'. Vol 21, No. 1, 2016

COSMIC

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



Effort vs COSMIC size is good for estimating



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

COSMIC

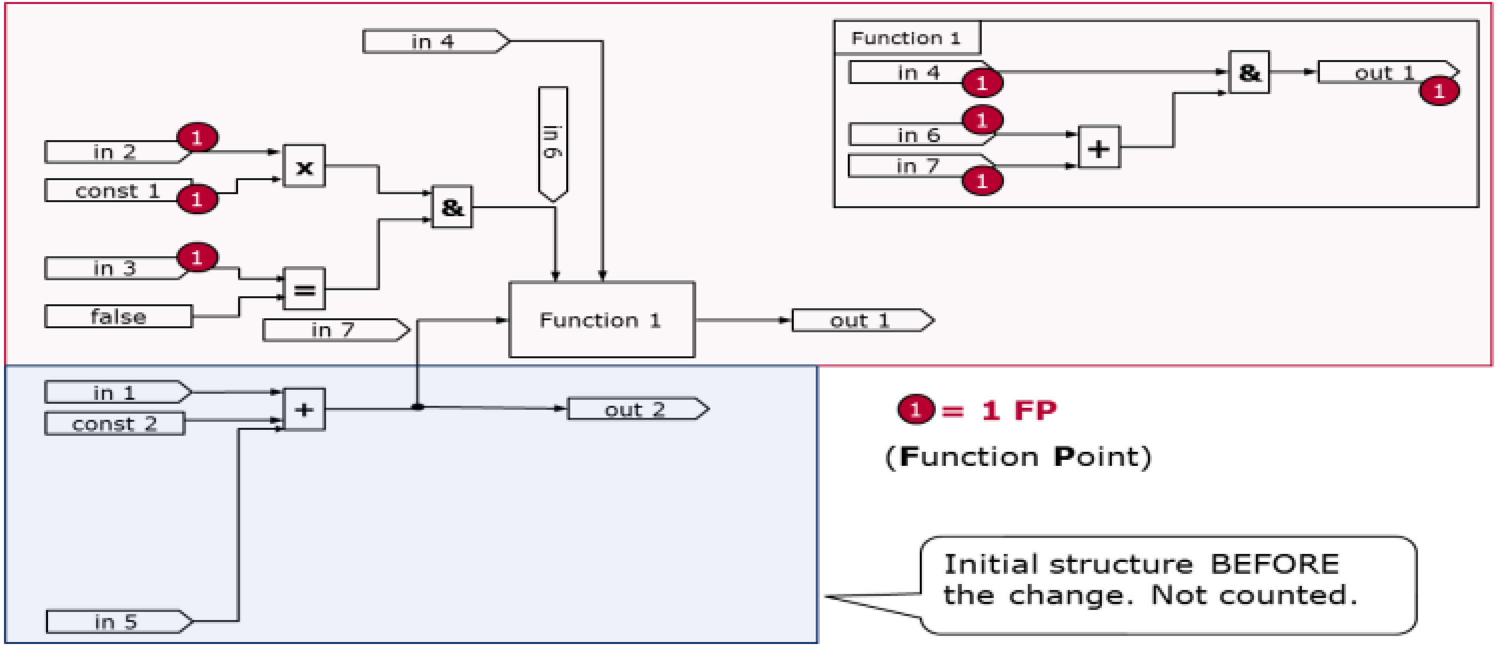
A User view of 'COSMIC for Agile'

- *"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

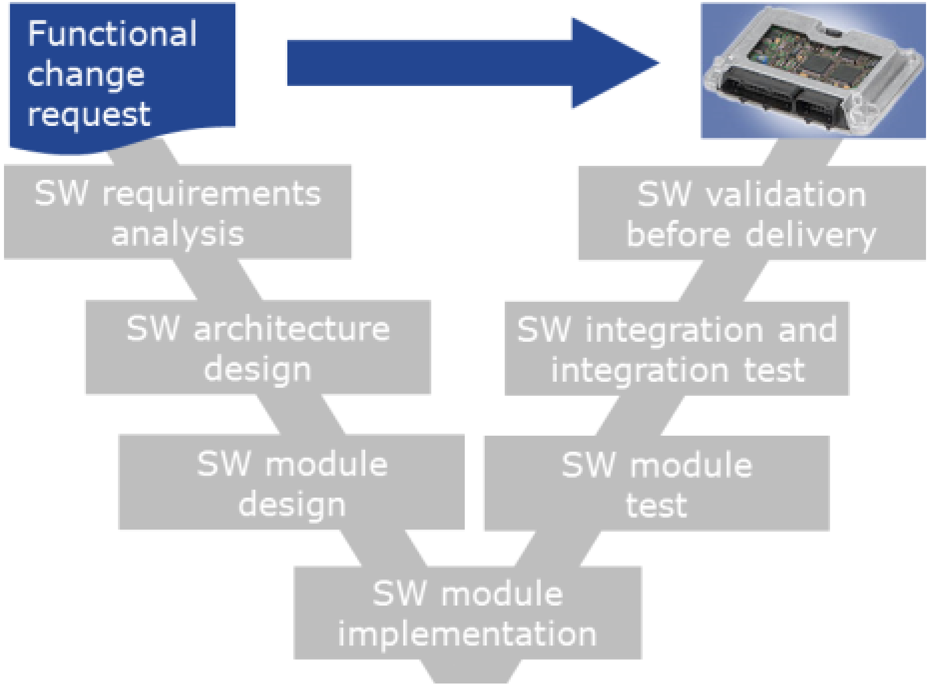
COSMIC

Case Study – Vector (Germany): Estimating Maintenance Effort



$Size_{FP}$ (Change) = size(added data) + size(modified data) + size(deleted data) = 7

Case Study – Vector (Germany): Maintenance Constraints



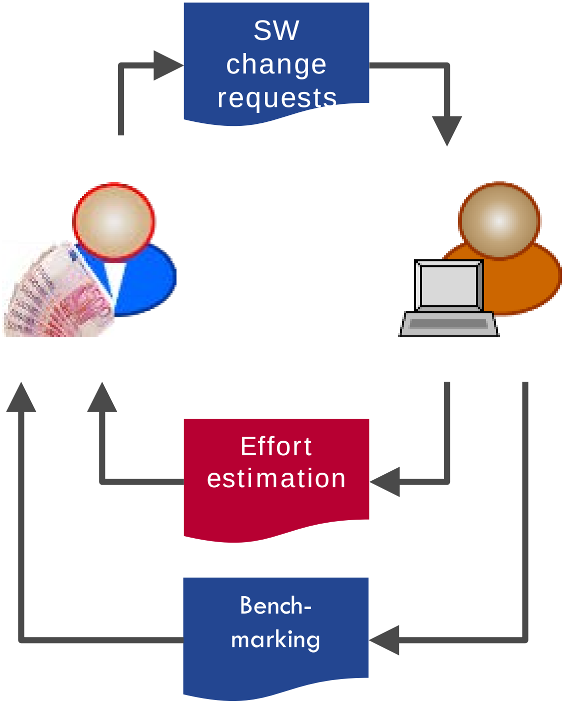
Requirements and design specification: Consistent level of documentation

Modeling: Same method, notation, semantics and visibility

Change documentation: All changes are covered and clearly marked

Case Study – Vector (Germany): COSMIC Benefits

- 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



COSMIC

Case Study – Vector (Germany): Results

- **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.

COSMIC

Agenda

- 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

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

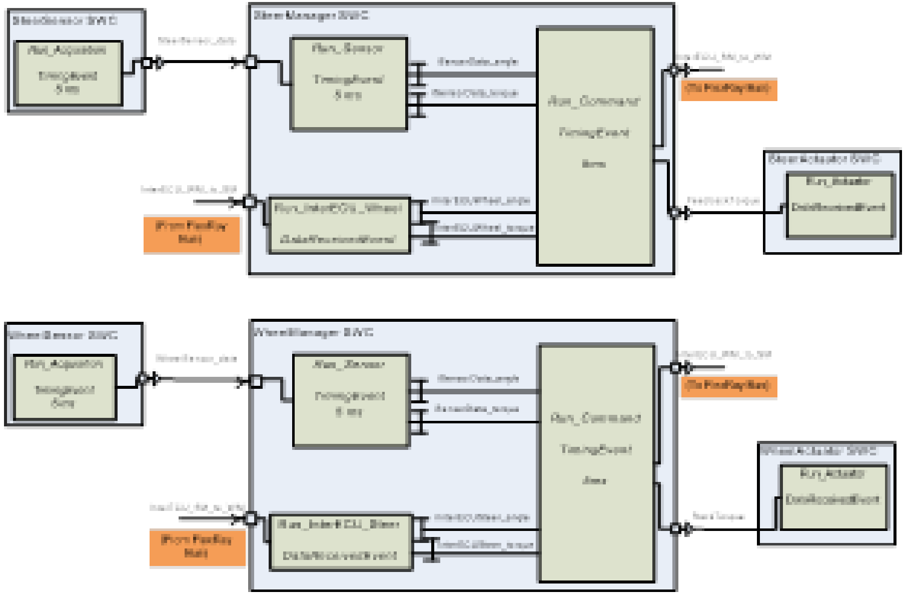
COSMIC Rules for Embedded Software
Requirements Expressed using Simulink®

ABSTRACT

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

Page 1 / 5
COSMIC rules

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



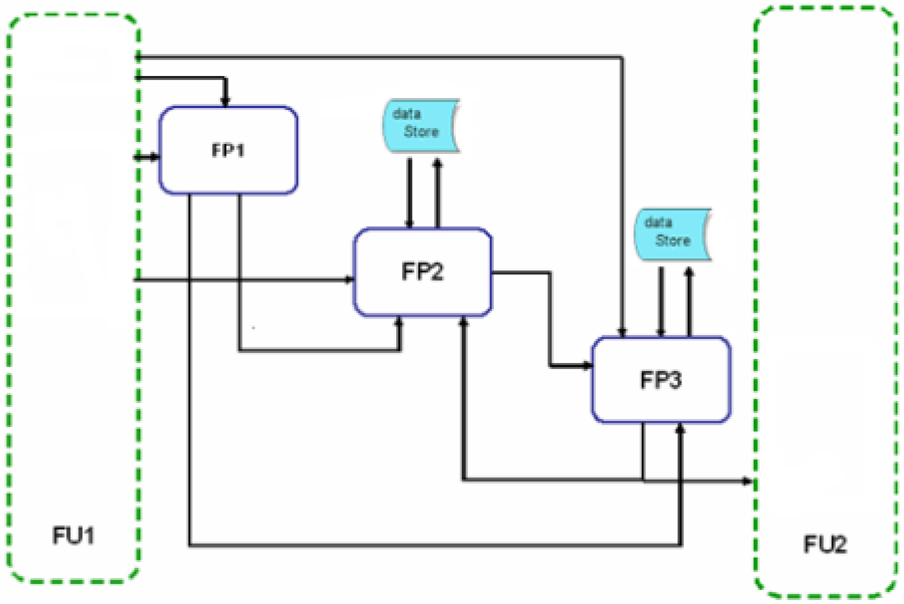
Runnables inside the software components -Steer-by-Wire system

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.

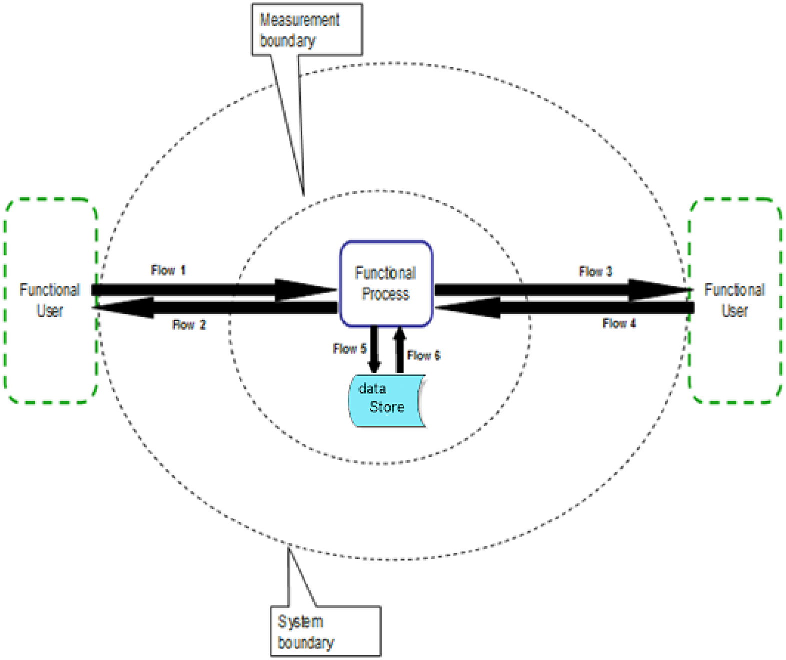
When Requirements are described with Graphs: Map to COSMIC

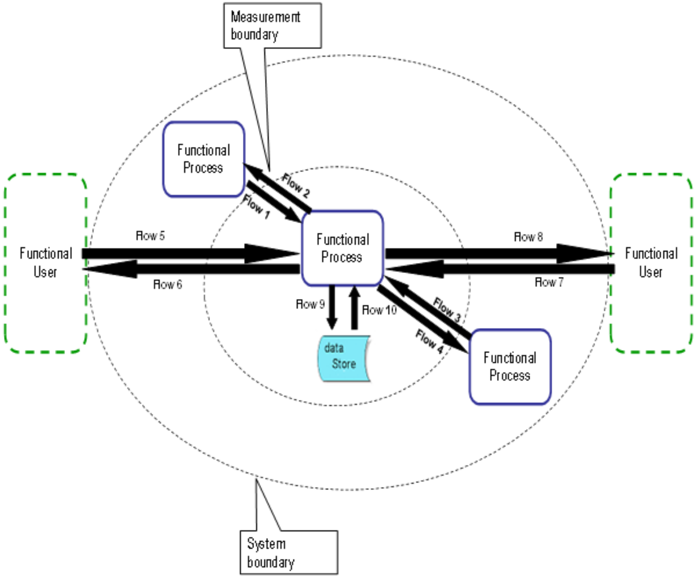
TABLE I.

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



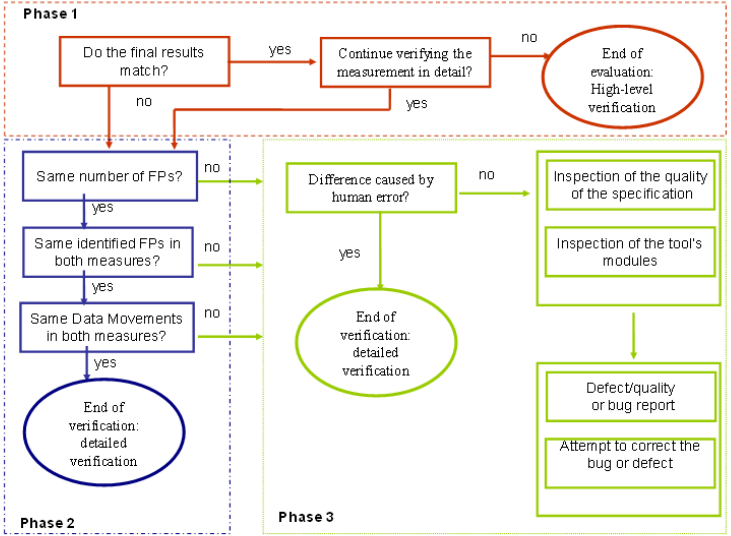
COSMIC representation of functional process (Single & Many) – For testing scenarios

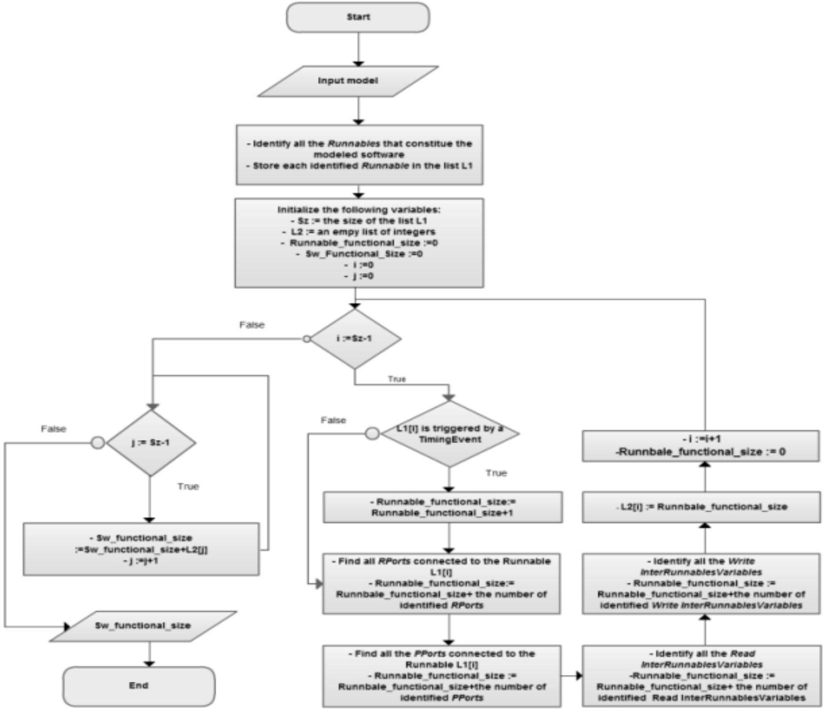




COSMIC

3-Phase Verification Protocal of Automation Accuracy





COSMIC

Automation Accuracy Reached with COSMIC

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**

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

Agenda

- 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

The COSMIC method is very widely used

- 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)

COSMIC

Summary of benefits

- Free, open
- Fundamental SE Principles: → future-proof, stable
- Very wide applicability
- Proven value for performance measurement & estimating
- ISO standard & GAO [1], NIST [2] endorsed
- 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

COSMIC

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