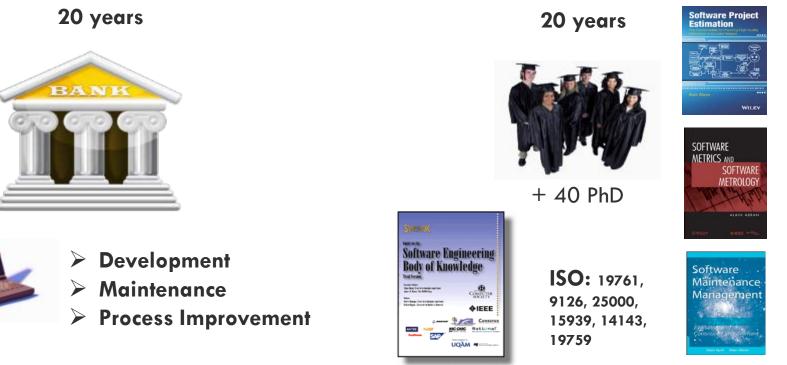


SOFTWARE PROJECTS ESTIMATION & CONTROL: VERSATILITY & CONTRIBUTIONS OF COSMIC FUNCTION POINTS

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



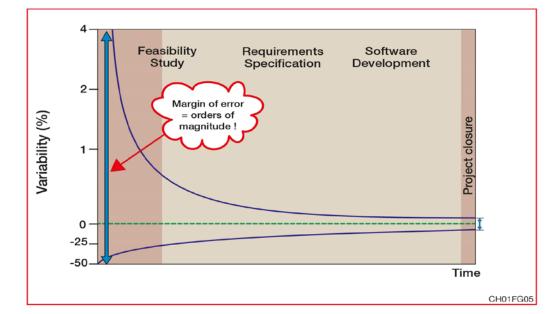
© Copyrights 2017: Alain Abran

ICEAA Bristol



- 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

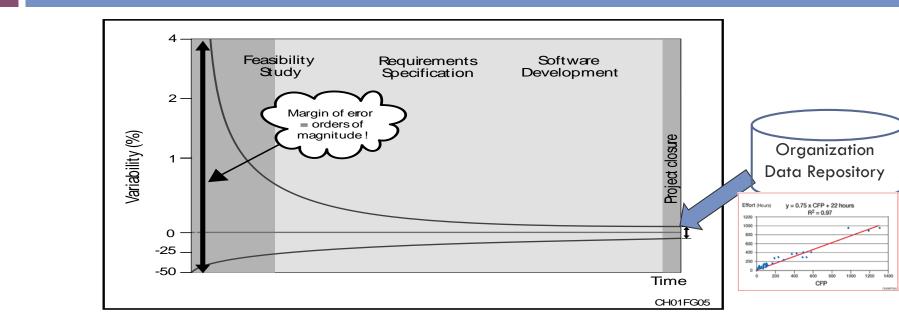
The Cone of Uncertainty across the Project Lifecycle



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

© Copyrights 2017: COSMIC and authors

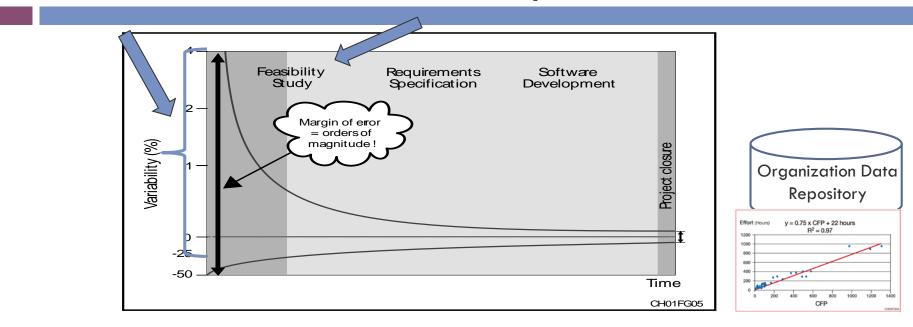
Productivity Models: Built with Data from Completed Objects



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

5

Estimation Foundations: Productivity Models with Uncertainties in the Inputs

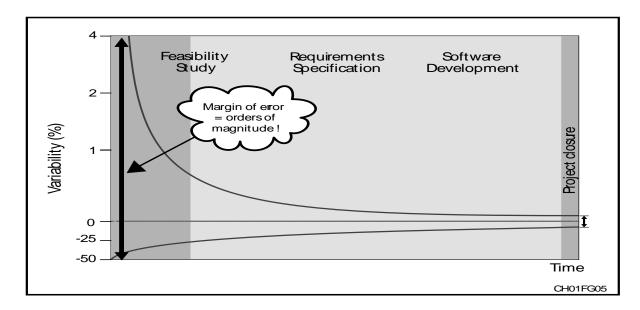


- > You do estimation upfront with a lot of uncertainty
- > What do you have upfront as available for estimation purposes?

Soltware:



What is Available & Measurable Across the Lifecycle?



© Copyrights 2017: COSMIC and authors

Software: Sizing: Options across: the Contemportand

Sizing method options:

8

- \succ Lines of code:
- Usecase Points,
 Object Points, ..
- Story Points

- X Can't estimate until software designed
- X Technology-dependent, no standards
- X Technology dependent, no standards, 🔤
- X Mathematical validity?
- X Entirely subjective



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



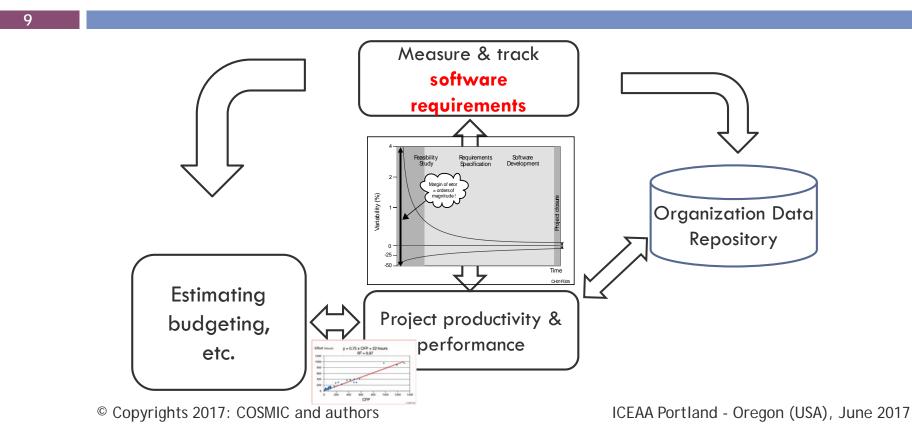
Technology-indepen



measurable software Requirements: the



Fresented at the 2017 ICEAA Professional Development & Training Workshop Foundations for Estimation & Benchmarking Studies





- 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



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

<u>Inputs - Matrix</u>

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

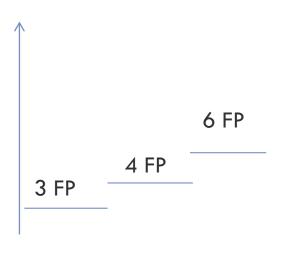
<u>Transactions: weights in</u> <u>FP (Function Points)</u>

© Copyrights 2017: COSMIC and authors

Presented at the 2017 ICEAA Professional Development & Training Workshop 1St Generation of Function Points: Step Functions

Function Points (FP)

12



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

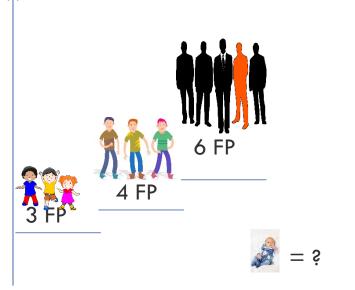
Presented at the 2017 ICEAA Professional Development & Training Workshop

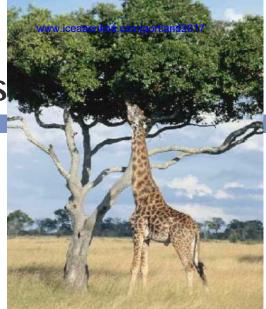


13

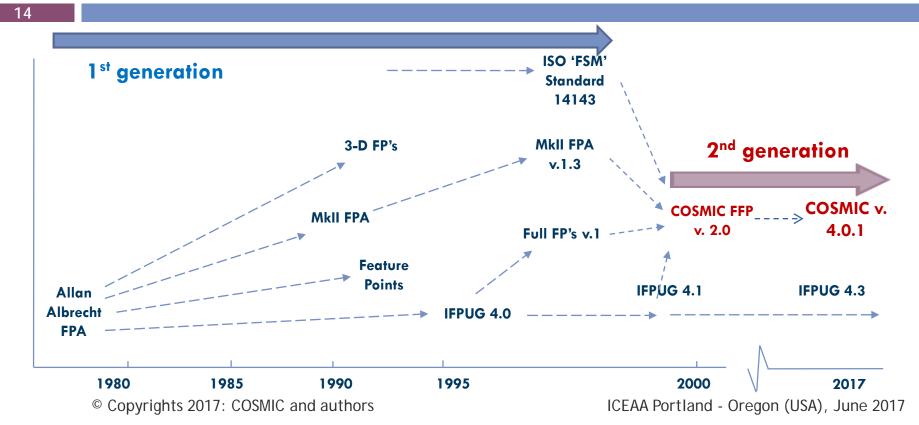
1st Generation of Function Points







1st & 2nd generation of Function Points





Every software is different, but what is common:

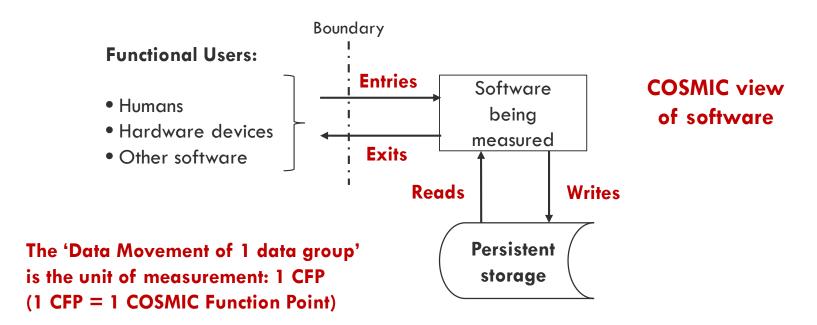
In all software?

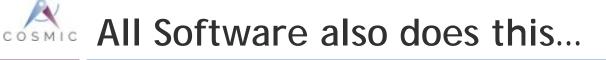
- In different types of software?
- In very small or extremely large software?

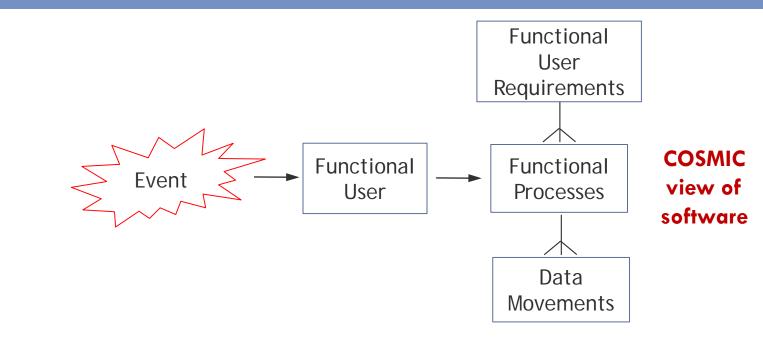
Собинс

16

2nd Generation of Function Points: All



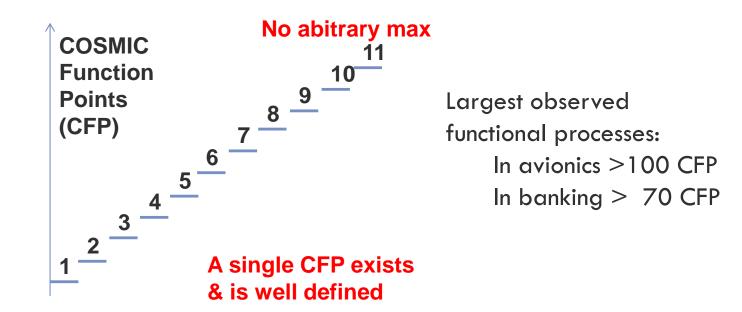




© Copyrights 2017: COSMIC and authors

17





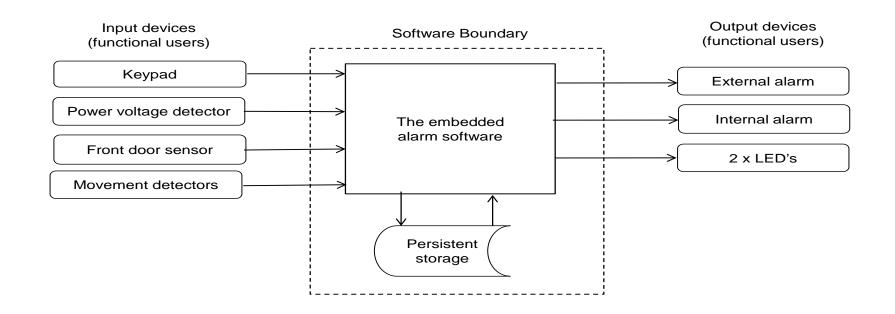
© Copyrights 2017: COSMIC and authors

www.iceaaonline.com/portland2017



Example 1: Intruder Alarm System -Requirements

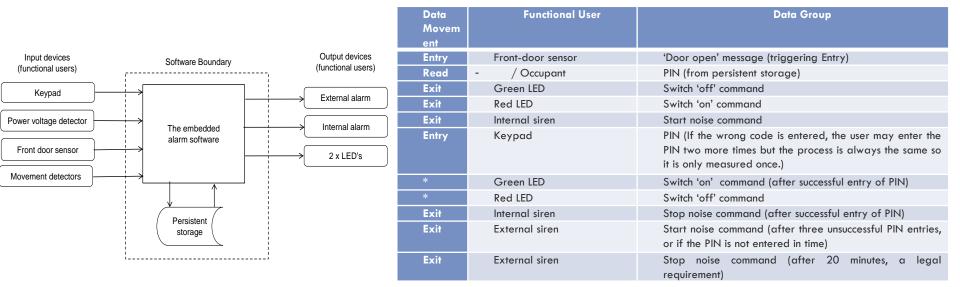
19_



Example 1: Intruder Alarm System - COSMIC size

Functional process: Possible intruder detected.

Triggering event: Door opens whilst alarm system is activated.



Size = 9 CFP (COSMIC Function Points)

© Copyrights 2017: COSMIC and authors

20



In summary: COSMIC Function Points

- 21
- Designed by an international group of software measurement experts
 - COSMIC: Common Software Measurement International Consortium
- To measure the <u>Functional User Requirements of</u>:
 - 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 <u>www.cosmic-sizing.org</u>)



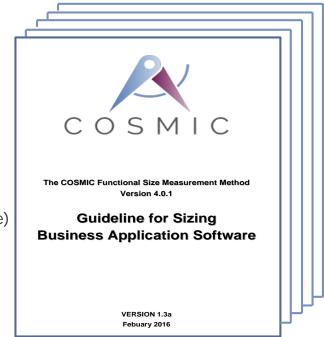


- 1. Software Effort Estimation & Software Size
- 2. COSMIC: 2nd generation of Function Points
- 3. Versatility of COSMIC Function Points
- 4. Contributions of COSMIC to Estimation Models
- 5. Early & Quick COSMIC sizing at Estimation Time
- 6. Automation of COSMIC Function Points
- 7. Summary

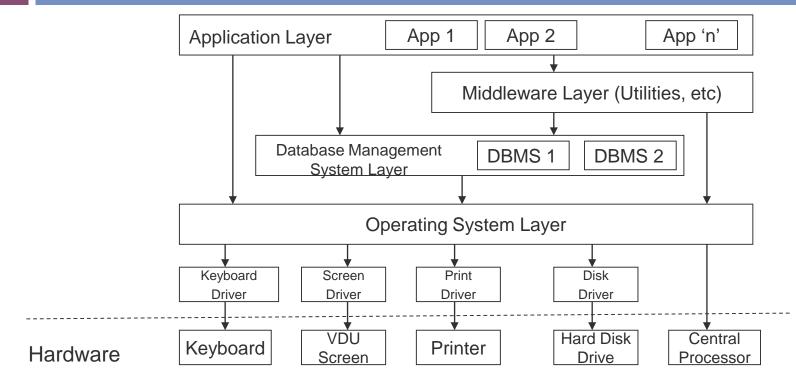


Versatility - Guidelines by Application Domains

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



Versatility - at any level of software requirements



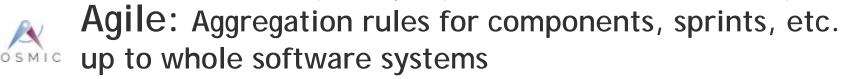
© Copyrights 2017: COSMIC and authors

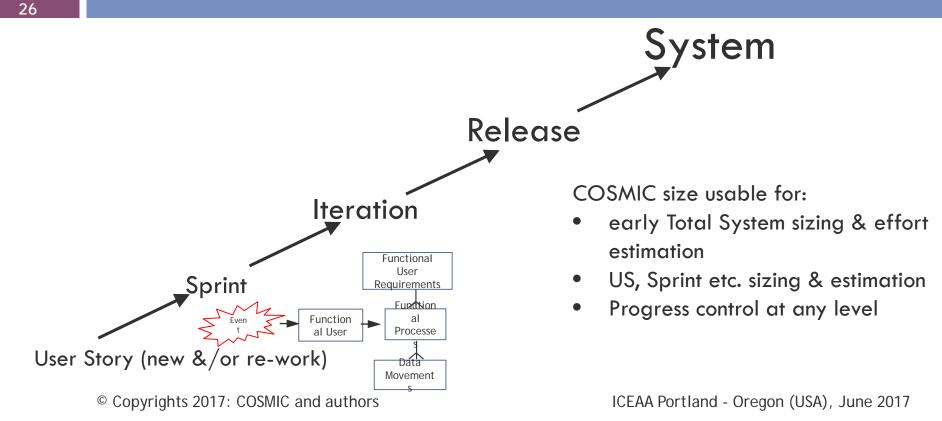
24

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.)







- 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



Practical experimentations with the COSMIC method in Automotive embedded software field

By: Sophie Stern

Renault

COSMIC Function Points

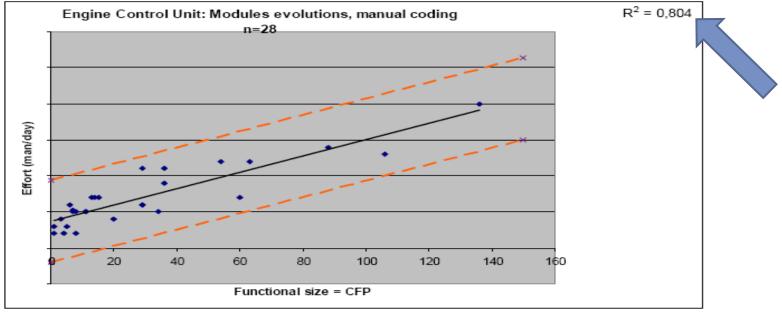
Theory and Advanced Practices





© Copyrights 2017: COSMIC and authors



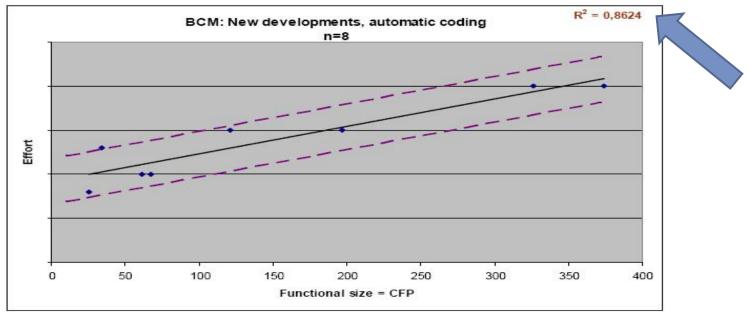


© Copyrights Renault 2012

© Copyrights 2017: COSMIC and authors

Data from Renault - 2012

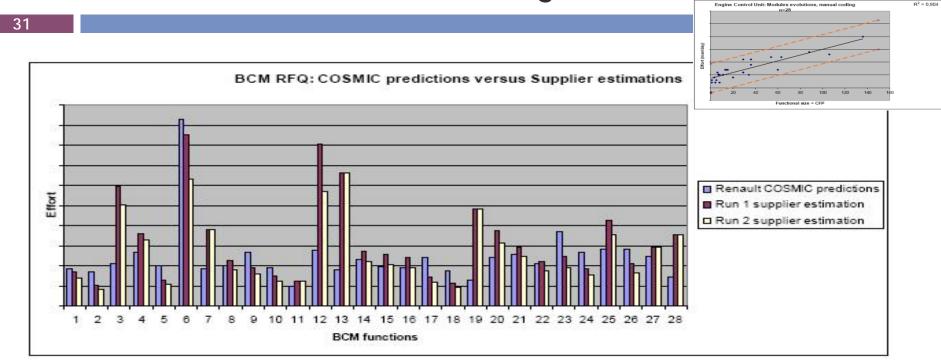
30



© Copyrights Renault 2012



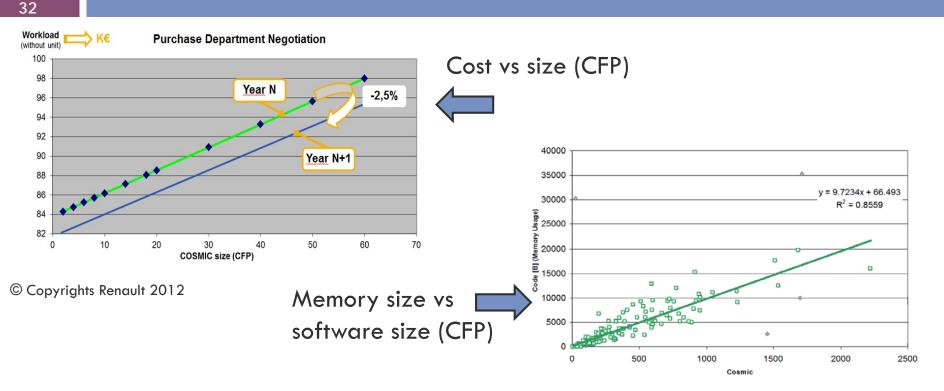
Renault: Estimation & Negociations



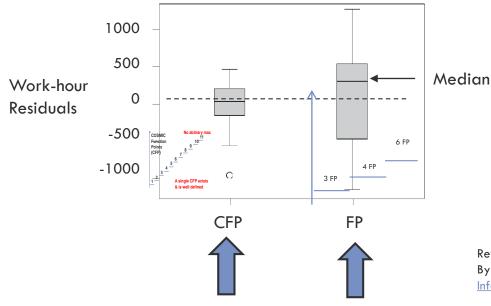
© Copyrights Renault 2012

COSMIC

Renault - Remarkable cost estimation accuracy from its ECU software specifications







© Copyrights 2017: COSMIC and authors

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



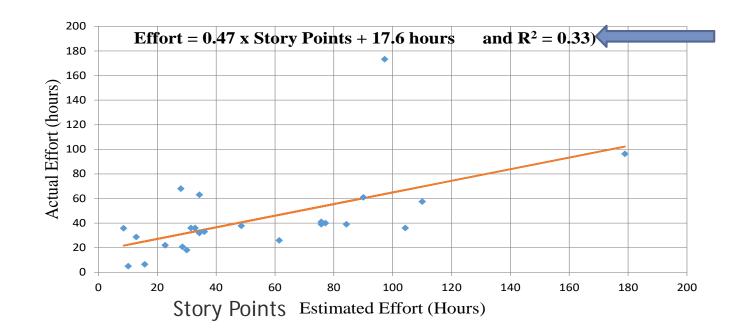
Industry Data - Example 3: Security & surveillance software systems

- 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
 - Fach task also measured in CFP

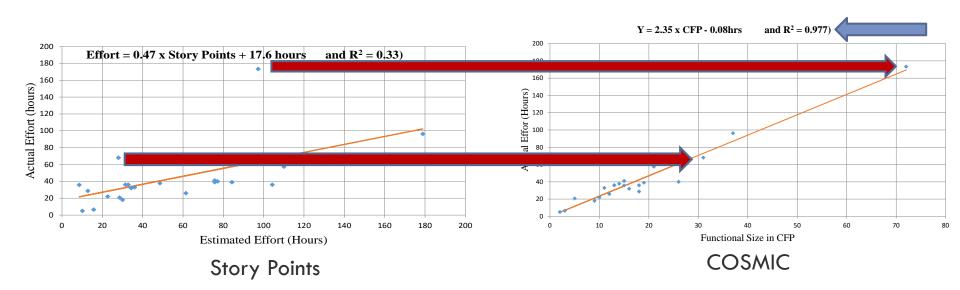
Ref. 'Effort Estimation with Story Points and COSMIC Function Points - An Industry Case Study', C. Commeyne, A. Abran, R. Djouab. Obtainable from www.cosmic-sizing.org 'Software Measurement News'. Vol 21, No. 1, 2016

© Copyrights 2017: COSMIC and authors

industry Data - Example 3: Security & surveillance software systems



Presented at the 2017 ICEAA Professional Development & Training Workshop Industry Data – Example 3: Security & surveillance software systems



36

Industry Data Example 4:



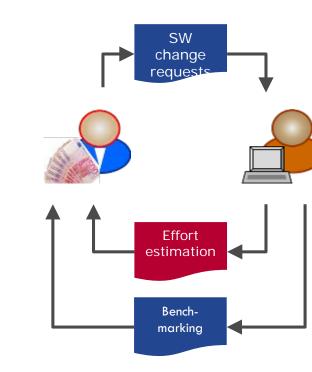
Vector Consulting Group (Germany) Manufacturing,

Engineering, Automotive, ..)

37

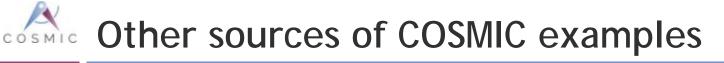
COSMIC Benefits

- 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

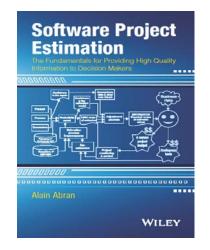


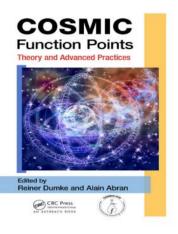
www.iceaaonline.com/portland2017

© Copyrights 2017: COSMIC and authors



COSMIC web site at: <u>www.cosmic-sizing.org</u>





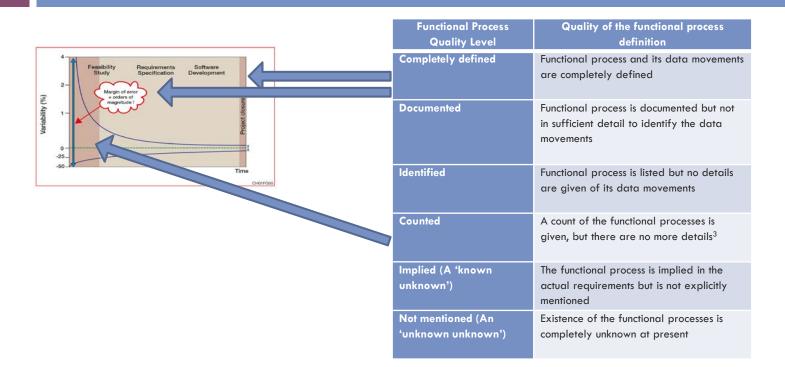
© Copyrights 2017: COSMIC and authors



- 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

Presented at the 2017 ICEAA Professional Development & Training Workshop Quality of the documentation of a functional process at measurement time

40

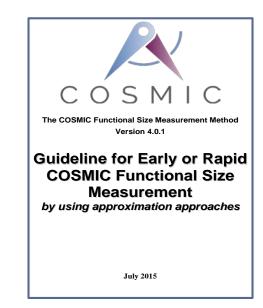




COSMIC Guidelines for Early or Rapid sizing

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





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
•••						



Example 2: Equal size bands

43

Equal size bands from 37 business applications

Band	.Average size of a	% of total	% of total number
	Functional Process	Functional Size	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	% of total	% of total number
	Functional Process	Functional Size	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

44

OSMIC

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 (1 <i>5</i> %)	10 (50%)	1 <i>5</i> (35%)	11.45	<50%
Complex FP	Little unknown	10 (10%)	15 (75%)	20 (1 <i>5%</i>)	15.25	>80%
Complex FP	Unknown (No FUR)	10 (1 <i>5</i> %)	18 (50%)	30 (35%)	21	<50%

© Copyrights 2017: COSMIC and authors



- 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

		Rev.: v 1.0
V		Date : 10 May 2012
HENRICO .	COSMIC Rules	
COSMIC	Rules for Embedded	Software
	nts Expressed using	
-		
This document presents the	ABSTRACT rules for measuring with COSMIC (IS	O 19761) the functional size of
software specifications docu	mented with Matlab-Simulink®.	

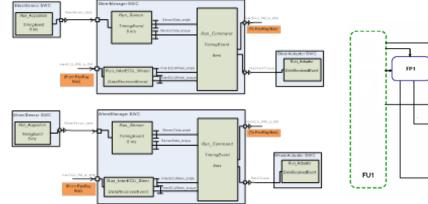
© Copyrights 2017: COSMIC and authors

46



Map the Graph Notation to COSMIC Model of Software

47_



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.

© Copyrights 2017: COSMIC and authors

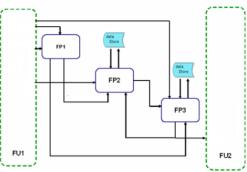


TABLE I.

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	1	Black arrow
Persistent storage		data Store	ISO 5807 stored data symbol in light blue



AUTOMATION ACCURACY REACHED WITH COSMIC

48

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.

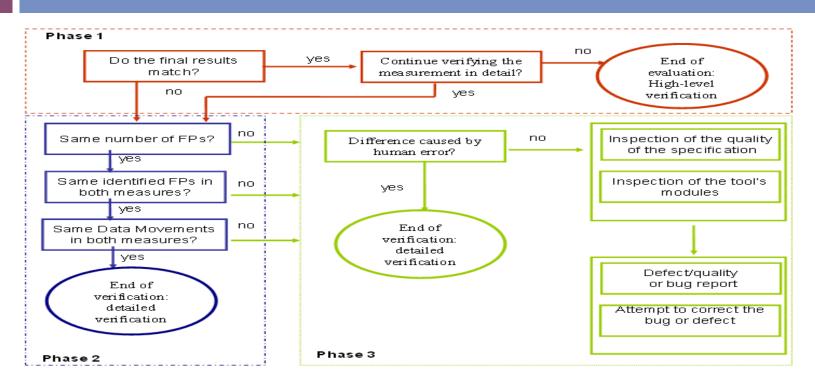
ICEAA Portland - Oregon (USA), June 2017

© Copyrights 2017: COSMIC and authors



Protocol for Verifying the Accuracy of Automation

49



© Copyrights 2017: COSMIC and authors



 Scade: A safety-certified language <u>https://www.youtube.com/watch?v=gjCvOjaC</u> Y88

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

© Copyrights 2017: COSMIC and authors



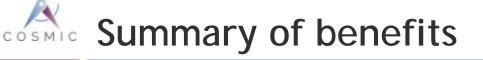
- 1. Software Effort Estimation & Software Size
- 2. COSMIC: 2nd Generation of Function Points
- 3. Versatility of COSMIC Function Points
- 4. Contributions of COSMIC to Estimation Models
- 5. Early & Quick COSMIC sizing at Estimation Time
- 6. Automation & Accuracy of COSMIC Function Points

7. Summary



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: GAO¹, NIST² documents
- + 600 certification exam holders (ex. China, India, Mexico, Italy, Poland, Turkey, Brazil)
- Two active forums (on Linkedin CUG, <u>www.cosmic-sizing.org</u>)
- 1) 'Cost Estimating and Assessment Guide' <u>http://www.gao.gov/new.items/d093sp.pdf</u>, March 2009
- 2) 'A Rational Foundation for Software Metrology', National Institute for Standards & Technology, NIST IR 8101, January 2016
 - © Copyrights 2017: COSMIC and authors



Free & open

53

- 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

© Copyrights 2017 COSMIC and authors



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



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

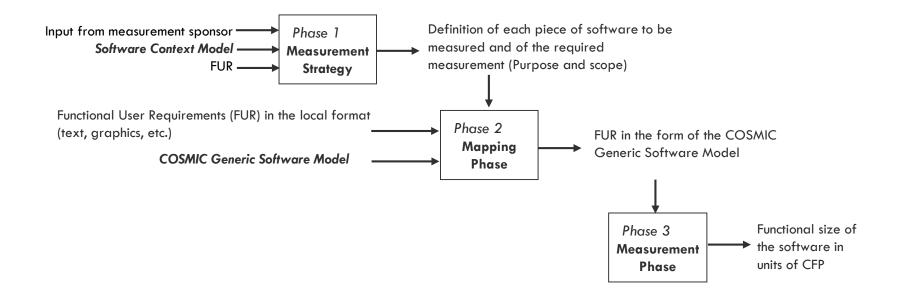
(www.cosmic-sizing.org)

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

© Copyrights 2017: COSMIC and authors

There is a well-defined Measurement²⁰¹⁷ Process

57



© Copyrights 2017: COSMIC and authors



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



Guideline on Non-Functional & Project Requirements

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

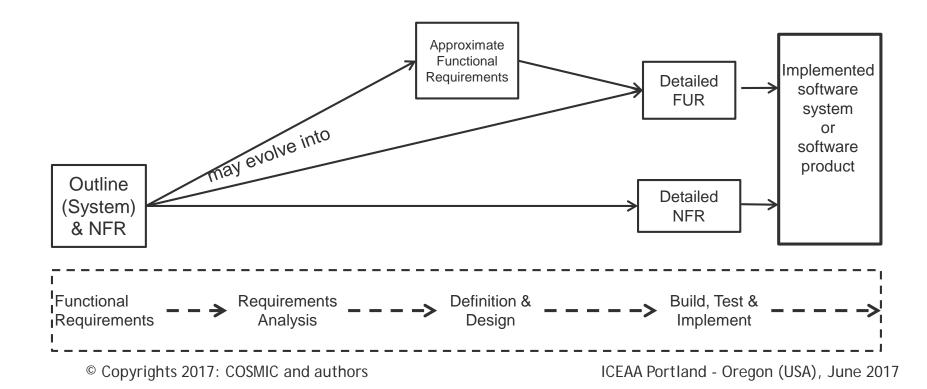
VERSION 1.0 September 2015 Version 1.

November 2015

© Copyrights 2017: COSMIC and authors



Abran & AI 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), Zaroa, Jordan



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-Saravreh. Software Engineering Department, Hashemite University (HU) 1100 Zarqa, Jordan./E-mail: khalidt@hu.edu.jo





A standards-based reference framework for system portability requirements

Alain Abran **, Khalid T. Al-Saravreh b, Juan J. Cuadrado-Gallezo **

⁴ Software Explorency and Information Including Departments Facilies de extensingle superiours (ITS), University of Quelter, Canada ³ Software Explorency, Existencies University (IRE) Jonán ⁴ Computer Cismon Departments (Intervity of Kold (April)

Article history: Broti well 4 A pril 2012 Broti well in real and Brom 4 Grapher 2012 Accepted 14 Networn her 2012 Accelative online 4 January 2012	In the system requirements place, the next-functional requirements (NR) are often optimal only provided a fairly high level, and they do not yet include the levels of detail measure for the system requirements a flace them any optimal flacestion at this in the handful either hyphe software net be lawbers or an appendix combination the two. The European ECSS write of shard afair of the arrequest instanty include perchain hyper- arised and any standard and the flacestimal at the second and the standard and mediate the two. The European ECSS write of shard afairs the arrequest instanty include perchain by requirements one of shards to prove from functional requirements (FW) for embedded and an ef-dires withous. A surviver
Keywands: Safrware Engineering Regularments Nan-functional regularments	postability-eduted concepts are dispersed throughout the ECSS, EE2-830, ESO 9126, ESO 24785, and E 2302-1 standards to describe at varying levels of detail, the variaus types of another postability angument at the system, software, and hardware levels. This paper organisas these dispersed postability concepts as
Porability requirement COSME-BIO 19761 ECE Sandards	term is to a standards basil of energia funerwork of poten prachibily regimeness. The availability of it framework on addition therearly/definition and up of it action of the system postability NNR and their detail about an aspectific postability functions to be based only the specifical about on the down or advance, opecific combustion of the two. The approach where do it is the research for the strature of the inference func- work is having on the post-it about for functions of the COMMC - DUPTO model, there where also were the strategies of the post-function of the strategies of the inference func- ation of the post-it about for functions and the the COMMC - DUPTO model, there also were the strategies of the post-it about the post-function of the transfer.

1. Introduction

In the system requirementsphase, the focus isoften on detailing and documenting the system functional requirements (FR) and their alloca-tion to the software and hardware parts of the system being designed. Non-functional requirements (NFR) play actifical mie in system development, including their use as selection criteria for choosing a mong alternative designs and ultimate implementations. NFR may also have a unsiderable impact on project effort, and should be taken into account for estimation numoses and when comparing project product/vities. Typically, these NFR are described 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 viewed, defined, interpreted, and evaluated differentlyby different people, particularly when they are stated briefly and vaguely [1-3], it is a challenge therefore to take NFR into account in software estimation and software benchmarking, 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 as quantitative inputs to an estimation pro-cess or to productivity benchmarking. In practice, requirements are initially addressed at the system level [6-3], either as high level system functional requirements (system-R) or as high level system NFR (system-NFR). Normally, such high-level

Conveponting author. E-mail address: skina brandwared.cs (A. Abran).

0920-5486/\$ - see front matter © 2013 Elsevier B.V. All rights reserved ng//dk.dolong/10.1016/5 ctl2012.11.003

functions, which may be implemented in hardware or software as software functional user requirements (software-FUR), for instance - see System-FR describe the functions required in a system, while asten NFR describe how those functions must behave in the astern [9,10]. In the software requirements engineering step, system-NFR may then be detailed and specified as software-FUR, to allow a software devel oper to develop, test, and configure the final deliverables to system users Functional requirements are the functions that the system finduding the software) is to offer, while NFR detail the manner in which those functionsare performed. FRare described using subject or predicate constructions (i.e. noun/verb), such as "The system can run on two or more kinds of devices or with two or more kinds of operating asterns." NFR are described using adverts or modifying clauses, such as: "The system can runnotwoor more kinds of devices, or with two or more kinds of op-

requirements must then be detailed and allocated to specifics related

erating geterns, that are easily or movemently transported." Within the European standards for the aerospace industry (ECSS) [11-15], ISO 9126 [16], IEEE-830 [17], ISO 24765 [18], and ISO 2382-1 [19], a number of mncepts are provided to describe various types of candidate portability require ments at the system, software, and had-ware kveb. However, these standards vary in their views, terminology, and portability coverage.

Currently, there exists no reference framework for the identification and spedification of system portability-NFR from the various views documented in international standards or in the literature.

Recent Advances on Electroscience and Compute

Model of Early Specifications of Performance Requirements at Functional Levels

Khalid T. Al-Sarayreh

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

Abstract- This paper presents an integrated standards-based model that help; 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

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

I. INTRODUCTION

I the system requirements at early development phases offered. 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 amplied 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 as a quantitative input to be measured and involved in the project budget estimation alongside with the FR [1-2]. Several approaches and methods are proposed from different Section 4 presents The Foundations of the proposed model of researcher's perspectives to facilitate dealing with these performance Requirements. A conclusion is presented in challenges; nevertheless, there is currently a lack of generic models for early addressing and measuring these requirements

This work was supported in part by the U.S. Department of Commerce ider Grant BS123456 (sponsor and financial support acknowledgment goes here). Paper titles should be written in uppercase and lowercase letters, not all uppercase. Avoid writing long formulas with subscripts in the title: short uppersance. Avoid writing iong Intrusias with subscripts in the title, short its firmula that indexify the releasests are fine (a.g., "A-Fo-B"). Do not write "flowing" in the title. Full names of authors are preferred in the author field, the are not required. Put a space between authors' initiality. F. A. Khalid T. Al-Starpred is now with the Hadsenite University, Prince Hassein Bia Advandilla II for Information Technology, Department of Schware Engineering, 13115 Zanga, Jordan. (e-mail: khalidtii/hu edu is)

Section 4 presents The Foundations of the proposed model of 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 systems and performance. The proposed framework includes the following performance conceptions, software performance

ISBN: 978-1-61804-290-3

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 remirements 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 The proper identification, specification and measurement of limitations viewed in the performance measures presently

The paper scope is to classify independently the all functionality allocated to software performance as a part of set pieces of the system application in the requirements phase for any software applications, whether the application has been built or it has already to be delivered.

In addition, the main contribution of this paper is the proposed model of software performance requirements. The proposed nonspecific model is considered as type of a orientation model in the common sense of an etalon standard that is being used for the measurement of software performance.

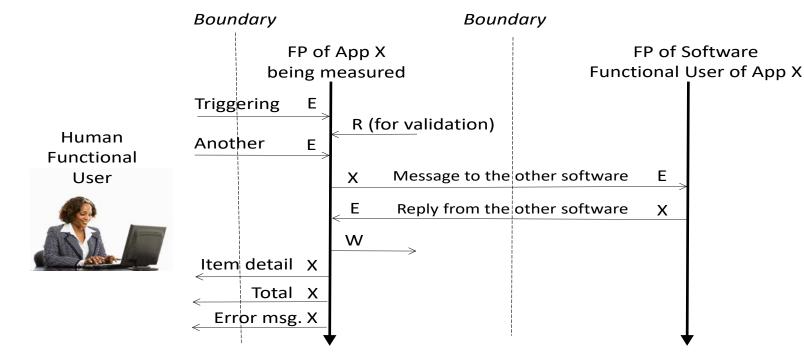
This paper is organized as follows. Section 2 presents the related works. Section 3 presents Performance REQUIREMENTS as defined in International Standards.

© Copyrights 2017: COSMIC and authors

236



Example 2: with a Message Sequence Diagram



Size = 9 CFP

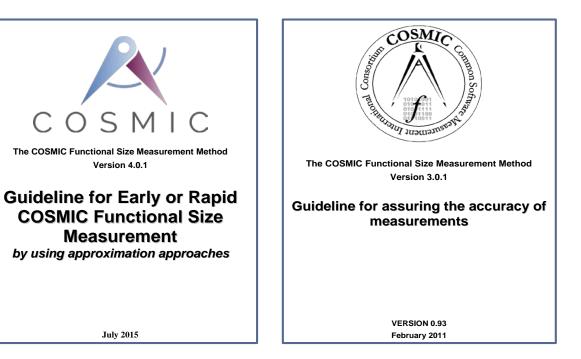
© Copyrights 2017: COSMIC and authors



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'





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	

<u>Inputs - Matrix</u>

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

2	AUC.	man	111811
Outp	ut & Er	nquirie	ès -
Sh	nared M	latrix	

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

Transactions: weights in <u>FP</u>

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

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

Files: weights in FP

<u>Files (internal &</u> <u>external) Matrix</u>

© Copyrights 2017: COSMIC and authors