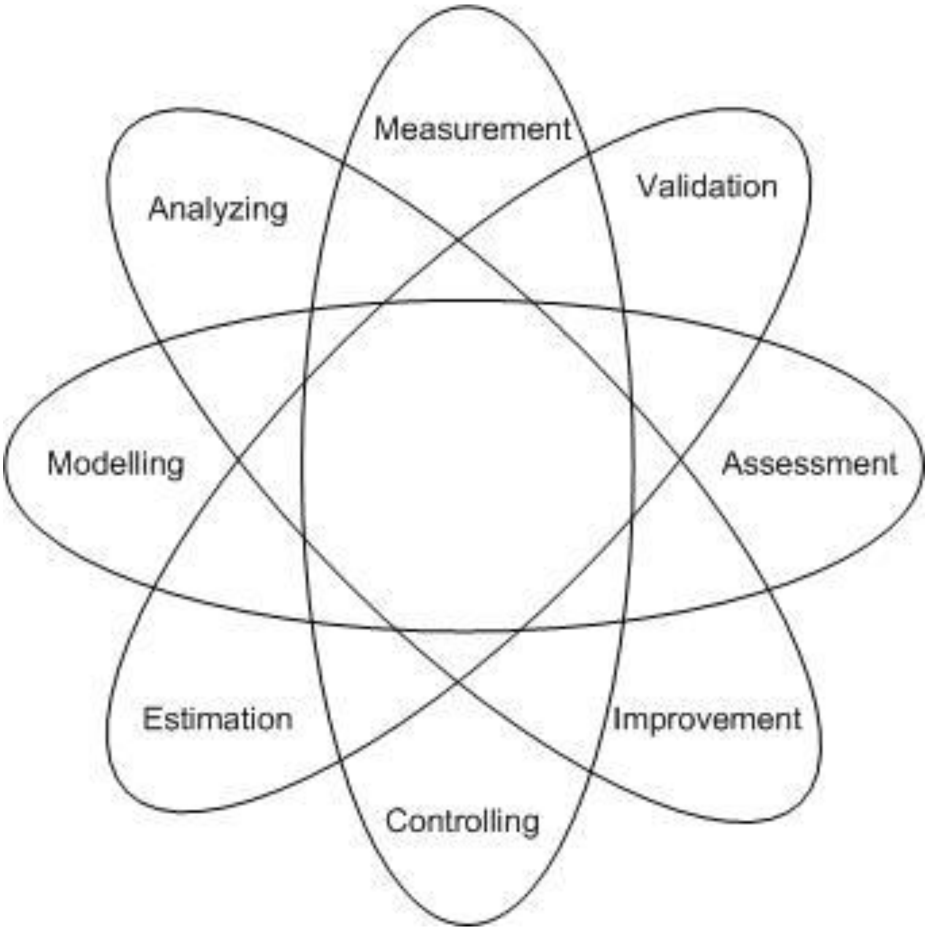
COSMIC

Volume 27, Number 2, October 2022

GI

Software Measurement News

Journal of the Software Measurement Community



Editors:

Alain Abran, Jens Heidrich, Reiner Dumke, Andreas Schmietendorf

SM L@b

Université du Québec
École de technologie supérieure

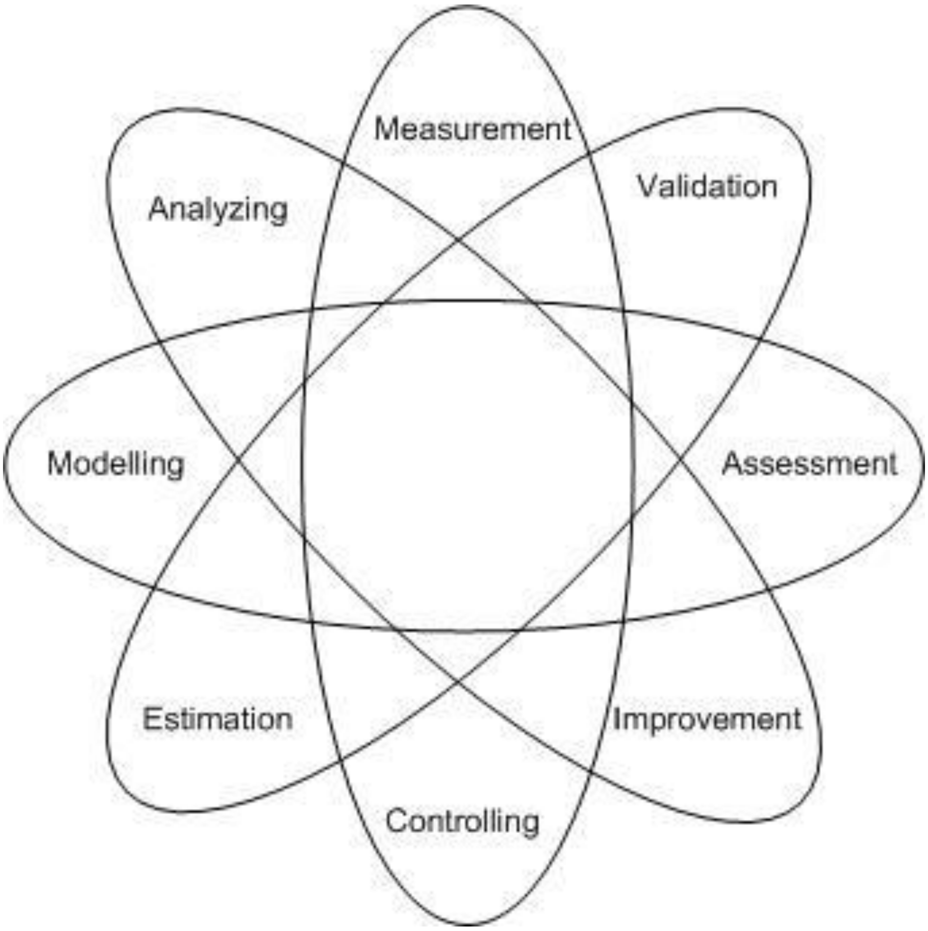
OTTO-VON-GUERICKE-UNIVERSITÄT MAGDEBURG

DASMA

Volume 27, Number 2, October 2022

Software Measurement News

Journal of the Software Measurement Community



Editors:

Alain Abran, Jens Heidrich, Reiner Dumke, Andreas Schmietendorf

CONTENTS

Editors:

Alain Abran
Professor and Director of the Research Lab. in Software Engineering Management
École de Technologie Supérieure, 1100 Notre-Dame Quest, Montréal, Quebec, H3C 1K3, Canada, `alain.abran@etsmtl.ca`

Jens Heidrich
Fraunhofer IESE
Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
`Jens.Heidrich@iese.fraunhofer.de`

Reiner Dumke
Professor on Software Engineering,
University of Magdeburg, Faculty of Informatics, Germany,
`reiner.dumke@t-online.de, http://www.smlab.de`

Andreas Schmietendorf
Hochschule für Wirtschaft und Recht
Alt-Friedrichsfelde 60 10315 Berlin, Germany,
`andreas.schmietendorf@hwr-berlin.de`

Editorial Office: University of Magdeburg, FIN, Postfach 4120, 39016 Magdeburg, Germany

Technical Editor: Dagmar Dörge

Vertrauenswürdigkeit von KI-Lösungen

(Implikationen im Data Science und Software-Engineering)

Veranstaltung der GI-Fachgruppe "Measurement & Data Science" (FG 2.1.10) im Rahmen der ESAPI-Community

15. November 2022 (Gastgeber: Fraunhofer IESE Kaiserslautern)

Die Vorträge am Vormittag können via MS Teams auch virtuell verfolgt werden!

Motivation

Das Vertrauen in Anwendungen der künstlichen Intelligenz ist von multidimensionalen Aspekten abhängig. Die „Ethics Guidelines for Trustworthy AI“ der Europäischen Kommission definieren verschiedene Prinzipien und Handlungsempfehlungen, wie das Abwenden von Schaden, Fairness oder transparente Prozesse, als Grundlage für Vertrauenswürdigkeit. Eine ausschließliche Berücksichtigung der technischen Eigenschaften entwickelter Lösungen, die sich z.B. an der ISO 25000 orientiert, ist zwar sinnvoll, reicht aber zur Gewährleistung vertrauenswürdiger KI-Lösungen nicht aus. Die VDE-Anwendungsregel VDE-AR-E 2842-61 des DKE-Arbeitskreises 801.0.8 bricht Vertrauenswürdigkeit in einzelne Qualitätsaspekte herunter, wie Zuverlässigkeit, Verfügbarkeit, Wartbarkeit, Funktionale Sicherheit, Cybersicherheit, Privatsphäre, Benutzerfreundlichkeit, Ethik/Moral und Robustheit. Mit Hilfe von KI-Lösungen gewonnene Klassifizierungen, Prognosen oder auch Bild-, Audio- und Videoanalysen implizieren Bedürfnisse hinsichtlich der Erklär-, Interpretier- und Reproduzierbarkeit. Dabei geht es nicht zuletzt um die Vermeidung diskriminierender Ergebnisse eingesetzter KI-Algorithmen. Die Reproduzierbarkeit erzielter Analyseergebnisse wird durch das BSI als direkte Vorrausetzung für die Verbreitung vertrauenswürdiger KI-Ansätzen genannt[1]:

> „Furthermore, reproducibility is a requirement for establishing causality for the interpretation of model results and building of trust towards the overwhelming expansion of AI systems applications.“ (Quelle des Zitats: BSI 2022)

[1] Quelle: Deep Learning Reproducibility and Explainable AI (XAI) Results of BSI's project research, Federal Office for Information Security 2022 https://www.bsi.bund.de, letzter Zugriff 13. September 2022

Unter Berücksichtigung der aufgezeigten Komplexität des Begriffs der Vertrauenswürdigkeit im KI-Diskurs bedarf es dennoch einfach zu handhabender Prinzipien und Methoden, die eine Auseinandersetzung mit sinnfälligen KI-Lösungen von vornherein nicht obsolet machen. Folgende Themenbereiche dienen der Anregung für potentielle Beiträge, selbstverständlich sind weitere Aspekte im denkbar.

Anmeldung:

Anmeldungen zum Workshop bitte über Herrn Sandro Hartenstein (Sandro.Hartenstein@hwr-berlin.de) bzw. unter www.cecmg.de realisieren.

Webseite zum Workshop

Weitere Informationen:
https://fg-data-science.gi.de und
https://blog.hwr-berlin.de/schmietendorf

Paper Submission:
https://easychair.org/conferences/?conf=esapi2022

Agenda

09:30 – 10:00

Eröffnung und Motivation zum Thema

Prof. Dr. Andreas Schmietendorf – Initiator des Workshops

Dr. Jens Heidrich – Gastgeber am Fraunhofer IESE

Dr. Andreas Jedlitschka – Sprecher der GI-Fachgruppe 2.1.10

10:00 – 12:00

Eingeladene Gastvorträge

Dr. Gaby Gurczik

Referentin für Grundsätze Künstliche Intelligenz und Datenökonomie beim BMDV

Dr. Rasmus Adler

Leiter des Programms Autonome Systeme am Fraunhofer IESE

Mittagspause 12:00 Uhr bis 13:00 Uhr

13:00 – 14:30

Moderiertes World Cafe – potentielle Themenvorschläge

- Risiken wahrscheinlichkeitsbehafteter KI-Ergebnisse
- Test und Validation von KI-Ergebnissen
- Möglichkeiten für erklärbare KI-Lösungen

Kaffeepause 14:30 bis 15:00 Uhr

15:00 – 16:30

Poster-Session mit einführenden Impulsvorträgen (jeweils 5 min.)

Sandro Hartenstein (Vertrauen in KI-Web-APIs)

Julius Schinschke (Vertrauen Datenquellen)

Lukas Scholz (Explainable Artificial Intelligence)

Daniel Krohmer (KI Security Testing)

Lisa Jöckel (Testing AI Systems)

Dr. Michael Kläs (Safe AI)

Ende des Workshops ca. 17:00 Uhr

Bemerkung: Änderungen der Agenda vorbehalten!

Programmkomitee

S. Aier,
Universität St. Gallen

F. Balzer,
IBM Deutschland

M. Binzen,
DB Systel GmbH

E. Dimitrov,
T-Systems

R. Dumke,
Uni Magdeburg

J. Marx Gómez,
Uni Oldenburg

M. Bauer,
CECMG

J. Heidrich,
Fraunhofer IESE

A. Johannsen,
TH Brandenburg

S. Kosterski,
Toll Collect

M. Lother,
Robert Bosch GmbH

P. Mandl,
HS München

M. Mevius,
HTWG Konstanz

A. Fiegler,
Microsoft

F. Victor,
TH Köln

T. Wiedemann,
HTW Dresden

S. Schmidt,
Deutsche Bahn AG

A. Schmietendorf,
HWR Berlin

C. Wille,
TH Bingen

M. Wißotzki,
HS Wismar

A. Jedlitschka
Fraunhofer IESE
Kaiserslautern

F. Simon,
Zurich Insurance Group

G. Gurczik
BMVI

R. Zarnekow,
TU Berlin

Kontakt zur Initiative

Andreas Schmietendorf

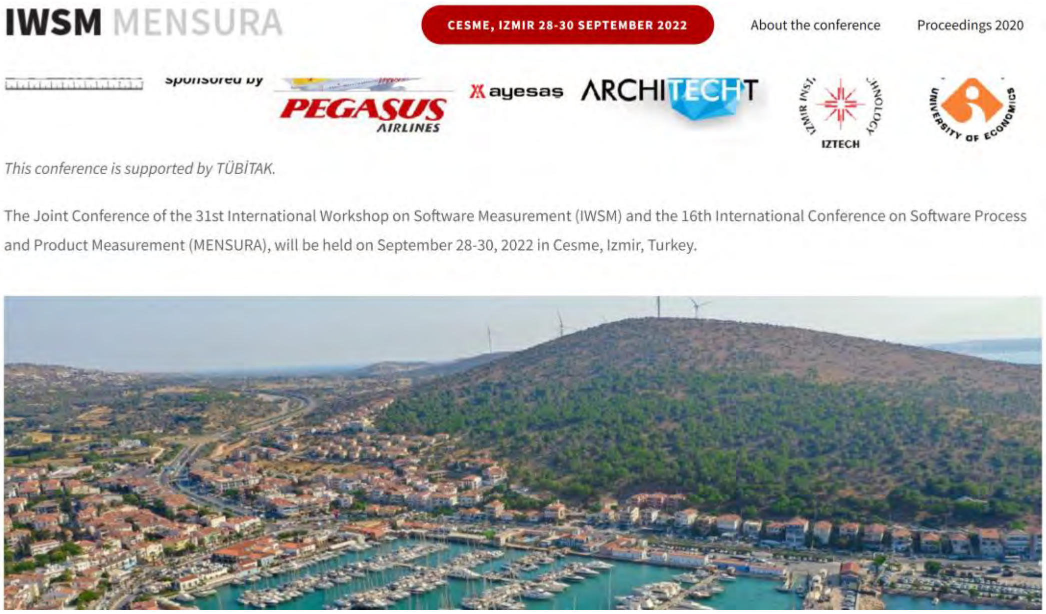
HWR Berlin - Berlin School of Economics and Law
E-Mail: andreas.schmietendorf@hwr-berlin.de

Jens Heidrich

Fraunhofer IESE Kaiserslautern:
E-Mail: Jens.Heidrich@iese.fraunhofer.de

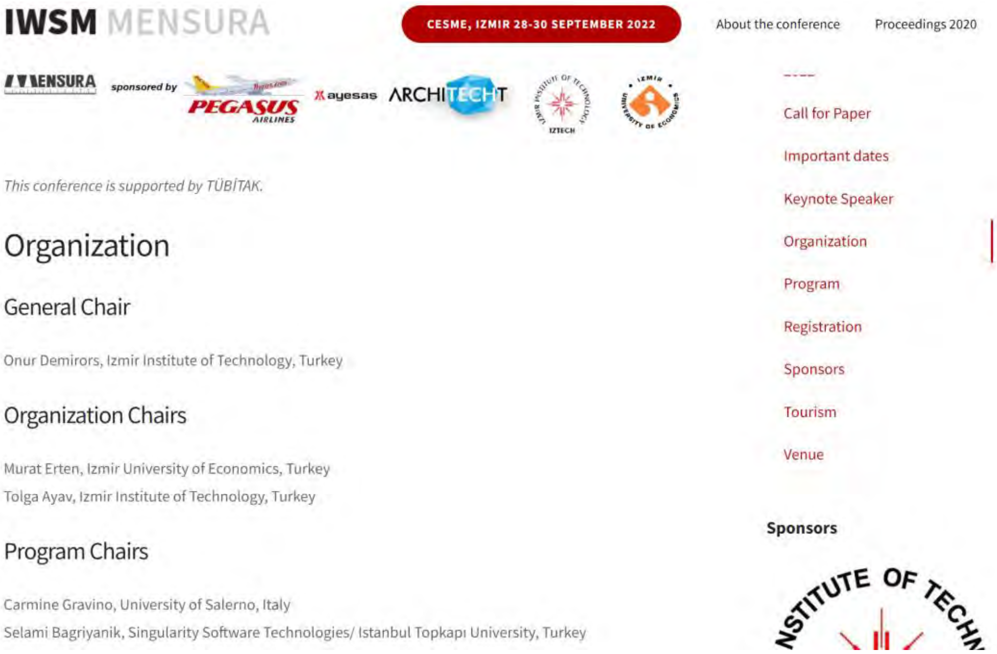
Report on the IWSM/Mensura 2022

September 28 – 30, 2022, Cesme, Izmir, Turkey



This conference is supported by TÜBİTAK.

The Joint Conference of the 31st International Workshop on Software Measurement (IWSM) and the 16th International Conference on Software Process and Product Measurement (MENSURA), will be held on September 28-30, 2022 in Cesme, Izmir, Turkey.



This conference is supported by TÜBİTAK.

Organization

General Chair

Onur Demirors, Izmir Institute of Technology, Turkey

Organization Chairs

Murat Erten, Izmir University of Economics, Turkey
Tolga Ayav, Izmir Institute of Technology, Turkey

Program Chairs

Carmine Gravino, University of Salerno, Italy
Selami Bagriyanik, Singularity Software Technologies/ Istanbul Topkapı University, Turkey

IWSM/MENSURA'22 Abstracts

Questionnaire Development for a Scientifically Founded Agile Assessment Model

Doruk Tuncel, Christian Körner and Reinhold Plösch

Agile software development methodologies have been a focal point of industry and academia over the past two decades. Organizations are interested in extracting the value of agile methodologies and increasing their business success, however, it is evident that merely applying agile practices does not necessarily make organizations agile. We are building a context agnostic agile assessment model based on agile values and principles to enable contextually appropriate self-assessments, and support organizations' agile transformation endeavors. In this paper, we share the results of our expert interviews with the focus on two principle pillars: Human Centricity and Technical Excellence. Results of this study show that the proposed assessment questionnaire appropriately addresses highly relevant aspects of agility. It is found practically useful by the experts, yet the coverage of human centricity targeted questions can be improved.

Maintenance Effort Estimation for Open Source Software: Current trends

Chaymae Miloudi, Laila Cheikhi, Alain Abran and Ali idri

Software maintenance of Open Source Software (OSS) has gained more attention in recent years and facilitated by the help of the Internet. Since volunteers in OSS do not record the effort of their contribution in maintenance tasks, researchers have to indirectly estimate the maintenance effort of such software. A review of the published OSS-MEE models has been performed using a set of 65 selected studies in a Systematic Mapping Study (SMS). This study analyses, discusses the state of the art about O-MEE and identifies trends through five additional Mapping Questions (MQs). In summary, various maintenance effort estimation (MEE) models were developed for OSS or industrial software. Researchers have mostly expressed the maintenance effort in terms of bug fixing, bug resolution time and severity in conjunction with bug report attributes. Regression Analysis and Bayesian Networks were most used estimation techniques, Recall, Precision, R2 and F-measure evaluation criteria in addition to k-fold cross validation method. Most of the models were implemented using WEKA, R software and MATLAB. More than half of the selected studies lacked of any validity analysis of their results. Trends are also discussed to identify a set of implications for researchers.

A Survey on COSMIC Students Estimation Challenge

Tuna Hacaloglu, Bilge Say , Huseyin Unlu, Neslihan Kucukates Omural, Onur Demirors

Software project management is a significant software engineering practice that is highly related to achieving software-specific project goals. This study aims to share students' perceptions of incorporating an international software estimation challenge called "COSMIC Students' Estimation Challenge" into a software project management course. For this aim, students were taught the COSMIC Functional Size Measurement method and entered the competition. After the competition, a questionnaire asking for the students' opinions was collected. The objective of the research is to get an insight into to what extent incorporating this type of competition activity -a challenge- can contribute to students' learning perceptions. In the long run, the findings can contribute to creating a foresight about making the necessary curriculum arrangements to form a more up-to-date and dynamic education plan by including the methods applied in the software industry in Software Engineering education. The results suggest that this kind of competition experience and preparation is helpful for students to learn the COSMIC method.

Automated COSMIC Function Points Measurement for C Program Using Regular Expressions

Donatien Koulla Moulla, Oumate, Ernest Mnkandla, Hassan Soubra and Alain Abran

Functional size measurement (FSM) is an important basis for measuring productivity and estimating the effort required for software activities. Automating FSM can be very valuable for organizations with a large number of projects to measure in a very short time, and there are several issues related to manual FSM, including measurement errors due to human measurers, the cost of measurement, measurers' subjective interpretations, and assumptions regarding the project scope. This paper presents an automated FSM for the C programming language, selected for being the most popular programming language from 1965 to 2020 and, as of January 1, 2022, in the second position after Python language. The FSM procedure is based on COSMIC ISO 19761, which is an FSM method designed based on metrology and software engineering principles. An automated measurement prototype tool was also introduced. The automated measurement tool can be useful for organizations and practitioners.

Tailoring Choices Made in Scrum Projects: A Systematic Literature Review

Özgün Özkan, Özden Özcan-Top

Among various Agile Software Development (ASD) methods, Scrum is one of the most widely adopted ones. The Scrum Guide gives a clear description of the Scrum practices, artifacts, and roles. However, due to various project characteristics such as team size, team distribution, project domain, technology, and requirement stability levels, Scrum practices need to be tailored. In this study, we performed a Systematic Literature Review to understand how software development companies adopt Scrum into their contexts, and explore the tailoring choices made in Scrum practices, roles and artifacts. The results of this SLR show that most organizations tailor the existing Scrum components by either not implementing a specific practice, role, or an artifact or modifying the duration limits of the practices. We specified that there are limited number of studies reporting successful Scrum adoptions. Most of the studies in our paper pool report that the tailored Scrum components cause difficulties in getting customer feedback on time, not understanding the sprint goal well enough, having an unordered, unclear product and sprint backlog, and an increase in project failures.

Towards Universal IoT Metrics Automation

Hassan Soubra

The Internet of Things (IoT) refers to a network of physical objects: embedded devices- also known as “things”- with sensors/actuators and connectivity (Internet) that enables them to exchange huge amounts of data. Metrics are an omnipotent means allowing better management of systems, Software and devices. This paper discusses the current state of IoT metrics and the underlying diversity in defining them. It also attempts to define a universal list of IoT metrics to cover most of the basic measurable attributes for any IoT device. A tool to visualize and automate IoT metrics’ readings using the MQTT protocol has also been implemented and is presented in this paper.

Functional Size Measurement Of Quantum Computers Software

Khaled Khattab, Hatem Elsayed and Hassan Soubra

Software measurement is an effective technique for project management. It helps engineers to apply engineering concepts to software development, providing a quantitative and objective foundation for process and technology decisions. Many measurement procedures based on international standards have been proposed to obtain the functional size of software. Some of the proposed procedures are automated to minimize measurement variance caused by individual interpretations. However, all of the proposed procedures are focused on ’classical computer’ Software, and none addressed Quantum Computer Software. Based on the COSMIC-ISO 19761, and with functional requirements implemented in Qiskit, this paper presents a functional size measurement (FSM) procedure for Quantum Computer Software. The mapping of essential concepts in both Qiskit and COSMIC, as well as the establishment of mapping rules for obtaining the information held in the Qiskit programs that is necessary for measurement, are the foundations of the FSM approach proposed in this paper. Consequently, this procedure provides the foundation for automating the measurement of Quantum Computer Software expressed in Qiskit.

Comparative Analysis of Estimation Sizing Approaches to Determine Their Suitability

Shashank Patil, Rashmi Sardesai, Carl Bideau and Yashowardhan Sowale

Software Estimation has been a complex topic across the IT Industry. For successful delivery of any software solution, it is important that it should be standing on strong foundation of estimation. One of the important aspects of project effort estimation is that it should enable competitive pricing of the projects without compromising on quality and schedule of the solution. There are several sizing techniques and estimation approaches prevalent in the industry today to carry out project effort estimations. However, there is no technique or approach which can be termed as “On size fits all”. Each one of the approaches has its own merits and demerits and hence, it’s a challenge for project managers to identify which one is best suitable for their project. In this paper we attempt to study and analyze 2 prominent Estimation Sizing techniques (approaches) and evaluate them on their critical success factors. Generally, the most common factors which establish usefulness of any Estimation Sizing technique are 3R&T, i.e.: Reliability, Repeatability, Reproducibility and Turnaround Time Out of these 4 factors, here we have chosen to study and analyze Repeatability and Reproducibility factors for 2 prominent estimation sizing techniques viz: Efforts Estimates derived using Relative Sizing Efforts Estimates derived using Absolute Sizing.

A size measurement method for Enterprise Applications

Neslihan Küçükateş Ömüral and Onur Demirörs

Enterprise Applications are known as one of the best practices of software reuse. They are complex applications, including most of the business processes. In this domain, size measurements and effort predictions are mostly performed in an ad-hoc fashion, and they frequently suffer from schedule and budget overruns. We developed a size measurement method for Enterprise Applications and explained this novel method in this paper. We categorized transactions as “unchanged”, “changed”, and “new” in this method. We defined a size measurement unit, Data Transaction Point (DTP), and measured size as DTP in these categories. We conducted a sample size measurement with a well-known business process to demonstrate the implementation of the method.

Towards a COSMIC FSM Programming Language Compiler

Youssef Attallah, Hassan Soubra

COSMIC ISO 19761 is considered a second generation of functional size measurement methods. It has met its primary design goal to be of practical value in software project performance measurement and in estimating activities, both in research and industry. The method’s partial and full automation -with very high accuracy results- helped in accelerating its adoption and widespread use. This paper presents a new approach for automatically and rapidly obtaining COSMIC functional size of Software requirements by introducing a new means to directly express and implement these requirements using the COSMIC vocabulary. An example is also presented as a proof of concept of the approach proposed.

Tools/frameworks that support development process of AI-based software: validations in white literature

Tugba Gurgen Erdogan, Haluk Altunel and Ayca Kolukısa Tarhan

Context: Artificial Intelligence (AI)-based software has gained increasing interest, especially in the last decade, due to advancements in underlying technologies and demands in varying business domains. With the proliferation to develop such software, there appears a need for developing methods and supporting tools/frameworks. Purpose: In this paper, we focus on tools/frameworks to automate AI-based software development process, from a holistic view. We review the scientific studies that were empirically validated and also evaluate their proposals with respect to basic characteristics including theme, research methods, types, domains, and a number of cases in empirical validations. Method: We elicit relevant studies (with the contribution type of tool or framework) from a larger set of primary studies identified by a systematic literature review on AI-based software development process. We select 14 primary studies in this context and analyze them with respect to the purposes of the proposals. Results: We review tools/frameworks that support AI based software development process under four headings: software system development process, the development process of fair software, model development process, and model deployment and operation processes. We observe that domains of empirical validation are diverse while the number of empirical cases applied for validation is limited. Also, only half of the primary studies provide links to their proposals as open-source, which is very important for the repeatability of the empirical validations.

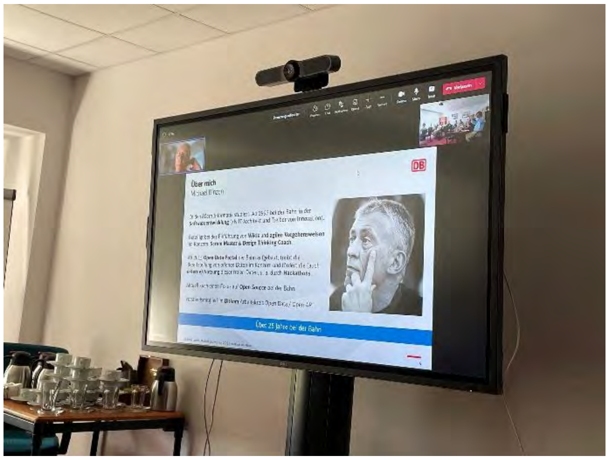
Impressionen des ganztägigen Workshops am 24.Juni 2022

Andreas Schmietendorf, HWR Berlin

Themenstellung: Verwendung vorgefertigter KI-Services im wissenschaftlichen und industriellen Diskurs (Möglichkeiten, Voraussetzungen und Grenzen)

Mehr als 20 Teilnehmer waren dem Aufruf zum Workshop an den Campus Berlin Lichtenberg der HWR Berlin gefolgt. Durch die Fach- und Impulsvorträge sowie interaktiv gestaltete World Cafés wurden aktuelle Themen rund um den Einsatz vorgefertigter KI-Services (u.a. vortrainierte Web-APIs für das maschinelle Lernen) aber auch die Möglichkeiten und Grenzen von Low-Code-Entwicklungsansätzen zur Diskussion gestellt.

M. Binzen, DB Systel, Bitkom e.V.

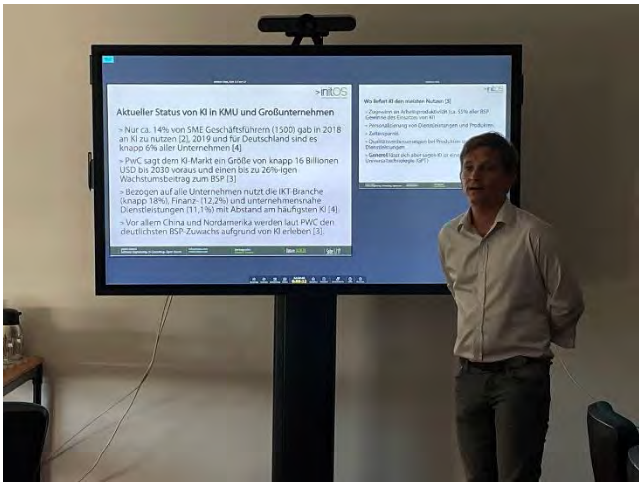


„Soll die KI als Werkzeug sinnfällig verwendet werden sind offene Schnittstellen zu Daten und Algorithmen bzw. Anwendungen unerlässlich“

Die Verdichtung des Bahnverkehrs im Diskurs des bestehenden Schienennetzes erfordern KI basierte Zuglaufprognosen, die allerdings mit mannigfaltigen Einflussaspekten (allg. Modellparameter) einhergehen.

Dr. J. Heidrich, Fraunhofer IESE

Für den Einsatz von KI-Lösungen in kritischen Infrastrukturen werden fundierte und risikogetriebene Engineerings-Prozesse benötigt. In diesem Zusammenhang wurde der Begriff eines Referenzmodells für vertrauenswürdige KI-Lösungen thematisiert. Dieses sollten das fachliche KI-Szenario und die damit verbundenen Mehrwerte, an Prototypen orientierte Implementierungsansätze aber auch Fragen der Qualitätssicherung in den Mittelpunkt des Prozesses stellen.



>initOS

Aktueller Status von KI in KMU und Großunternehmen

> Nur ca. 14% von SME Geschäftsführern (1500) gab in 2018 an KI zu nutzen [2], 2019 und für Deutschland sind es knapp 6% aller Unternehmen [4]

> PwC sagt dem KI-Markt ein Größe von knapp 16 Billionen USD bis 2030 voraus und einen bis zu 26%-igen Wachstumsbeitrag zum BSP [3]

> Bezogen auf alle Unternehmen nutzt die IKT-Branche (knapp 18%), Finanz- (12,2%) und unternehmensnahe Dienstleistungen (11,1%) mit Abstand am häufigsten KI [4]

> Vor allem China und Nordamerika werden laut PWC den deutlichsten BSP-Zuwachs aufgrund von KI erleben [3].

Wo liefert KI den meisten Nutzen [3]

> Personalisierung von Dienstleistungen und Produkten

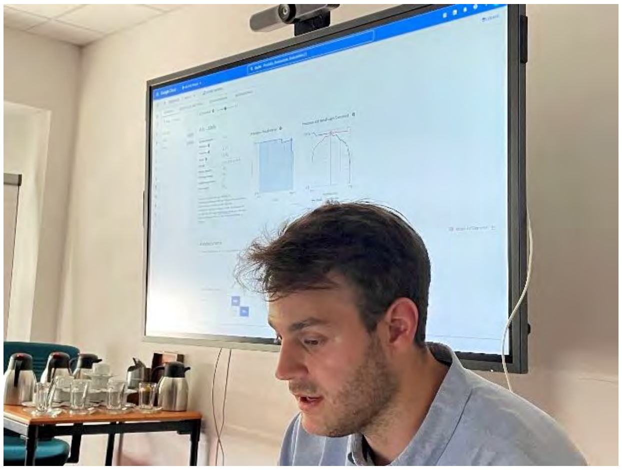
> Zeitersparnis

Dr. F. Kramer, initOS GmbH

Obwohl der Einsatz von KI-Lösungen bei KMUs mit vielen Vorteilen in Bezug auf z.B. die Agilität, Produktivität oder auch Personalisierung von Produkten und Dienstleistungen verbunden wird, kann die tatsächliche KI-Verwendung bei KMUs noch nicht überzeugen. Häufig fehlen klare Visionen und Ziele für KI-Anwendungen bzw. wirtschaftlich belastbare Anwendungsszenarien.

„Die Demokratisierung von KI bedeutet, dass KI nicht mehr ausschließlich ein Thema für Experten ist. Jetzt wollen Unternehmen die nächste Stufe erreichen, indem sie mehr Menschen KI zugänglich machen. Im Unternehmen können Kunden, Geschäftspartner, Führungskräfte, Vertriebsmitarbeiter, Produktionsmitarbeiter, Anwendungsentwickler und IT-Betriebsexperten Ziele für die Demokratisierung von KI sein."

Quelle: https://www.gartner.de/de/artikel/2-megatrends-dominieren-den-hype-cycle-von-gartnerfuer-kuenstliche-intelligenz-2020



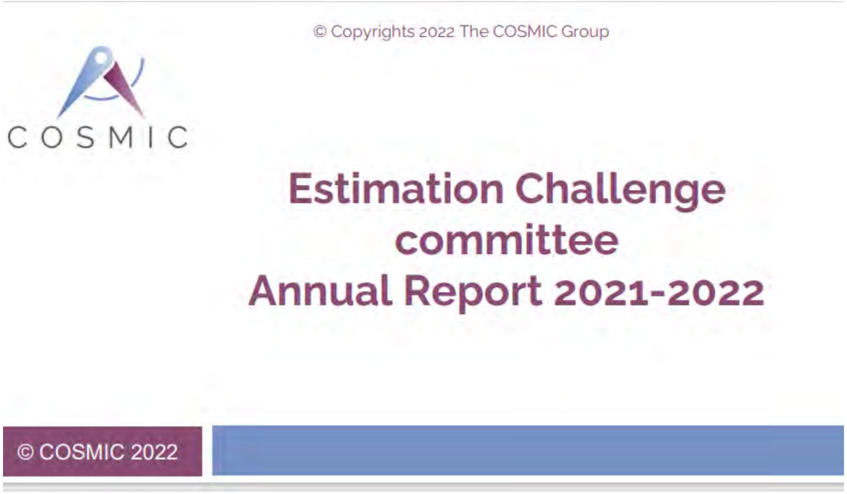
P.C. Kortus, HWR Berlin

KI-Anwendungen bedürfen der Berücksichtigung von Lernkurven und schnellen Feedbacks, was durch Low Code Ansätze unterstütz wird. Ein interessantes Beispiel findet sich mit der Entwicklungsplattform Vertext AI. Im Kern erfolgt der visuell orientierte Einsatz vorgefertigter KI-APIs für z.B. die Klassifikation oder auch Erkennung von Dokumenten und Bildern.

Für die Übernahme der Catering-Kosten bedanken wir uns ganz herzlich bei der ceCMG!

Software Cost Estimation Summit 2022

Alain Abran, ETS Montreal

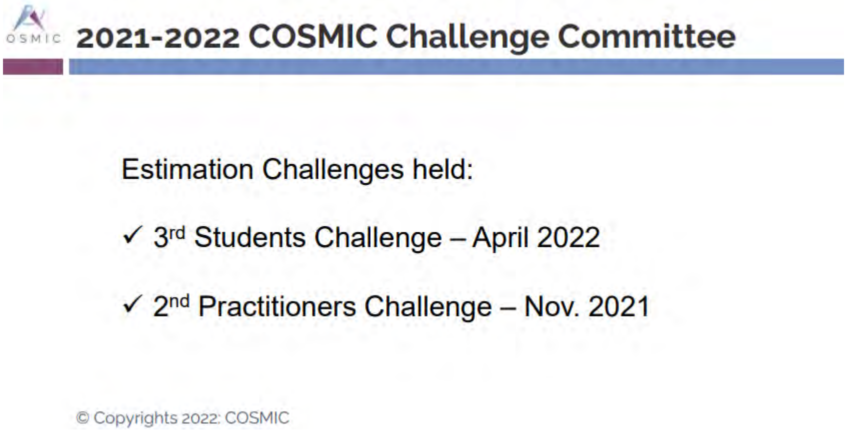


© Copyrights 2022 The COSMIC Group

COSMIC

Estimation Challenge committee Annual Report 2021-2022

© COSMIC 2022

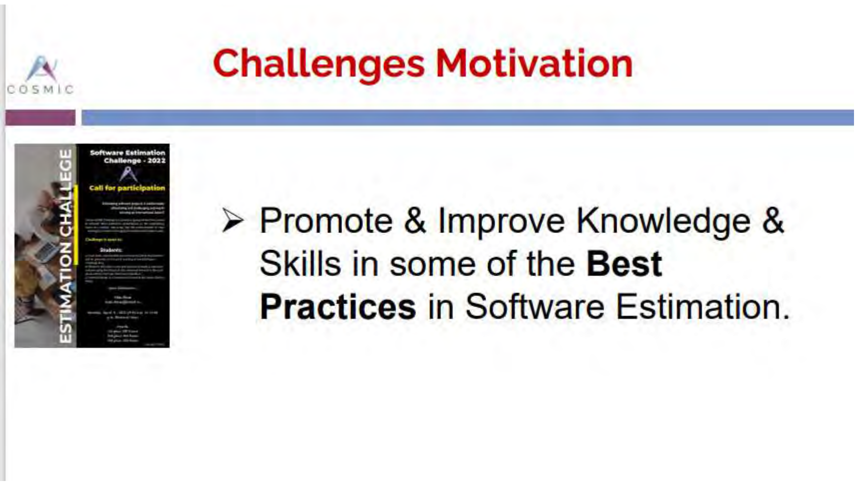


2021-2022 COSMIC Challenge Committee

Estimation Challenges held:

- ✓ 3rd Students Challenge – April 2022
- ✓ 2nd Practitioners Challenge – Nov. 2021

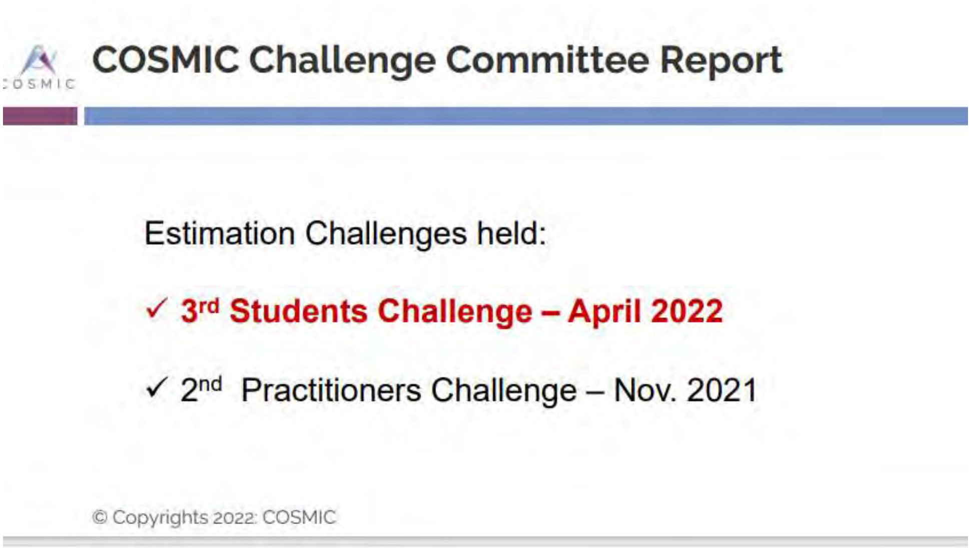
© Copyrights 2022: COSMIC



Challenges Motivation

ESTIMATION CHALLENGE
Software Estimation Challenge - 2022
Call for participation

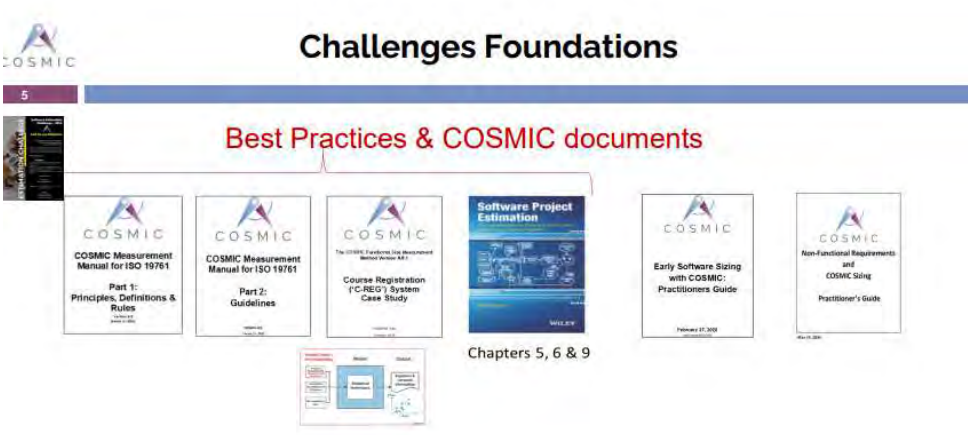
- ➢ Promote & Improve Knowledge & Skills in some of the **Best Practices** in Software Estimation.



COSMIC Challenge Committee Report

Estimation Challenges held:

- ✓ **3rd Students Challenge – April 2022**
- ✓ 2nd Practitioners Challenge – Nov. 2021

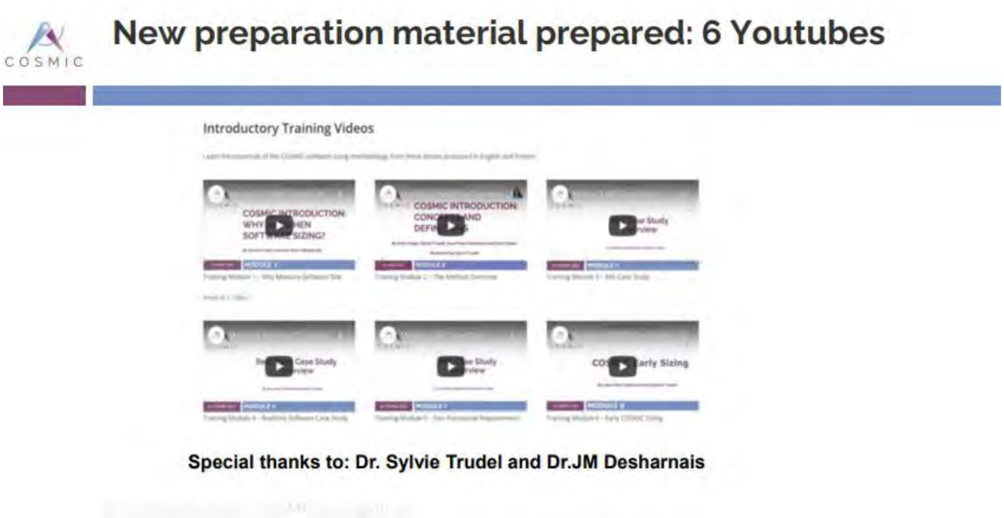


Challenges Foundations

Best Practices & COSMIC documents

5
COSMIC
COSMIC Measurement Manual for ISO 19761
Part 1: Principles, Definitions & Rules
COSMIC
COSMIC Measurement Manual for ISO 19761
Part 2: Guidelines
COSMIC
Course Registration ('C-REG') System Case Study
Software Project Estimation
WILEY
COSMIC
Early Software Sizing with COSMIC: Practitioners Guide
COSMIC
Non-Functional Requirements and COSMIC Sizing
Practitioner's Guide

Chapters 5, 6 & 9

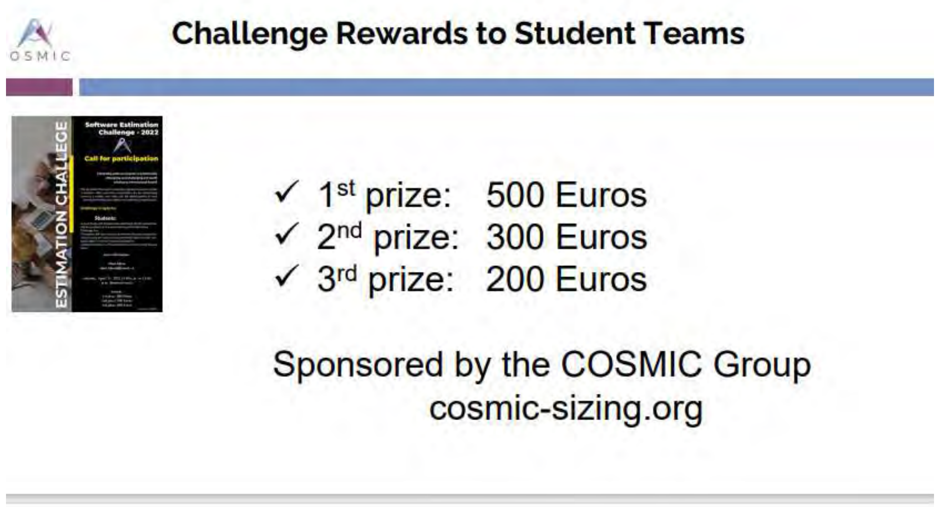


New preparation material prepared: 6 Youtubes

Introductory Training Videos

COSMIC INTRODUCTION: WHY [illegible] SOFTWARE SIZING?
COSMIC INTRODUCTION: CONCEPTS AND DEFINITIONS
Case Study Overview
Case Study Overview
Case Study Overview
COSMIC Early Sizing

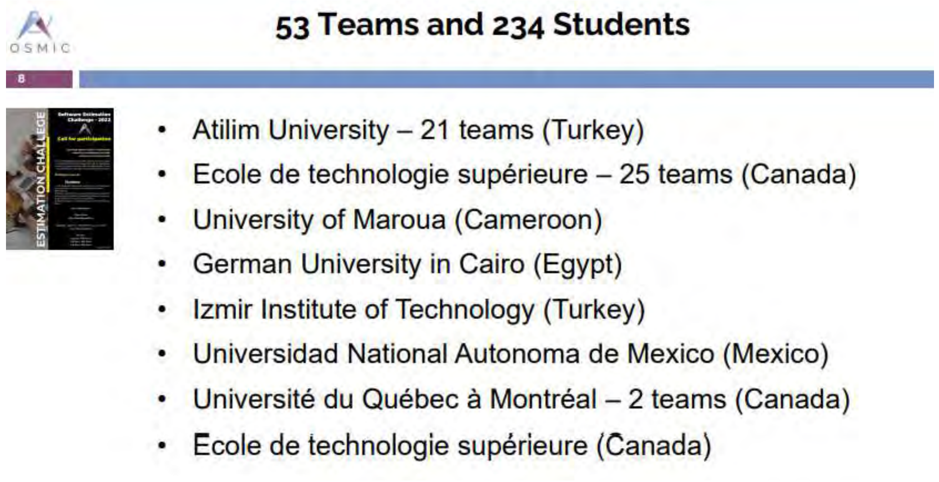
Special thanks to: Dr. Sylvie Trudel and Dr.JM Desharnais



Challenge Rewards to Student Teams

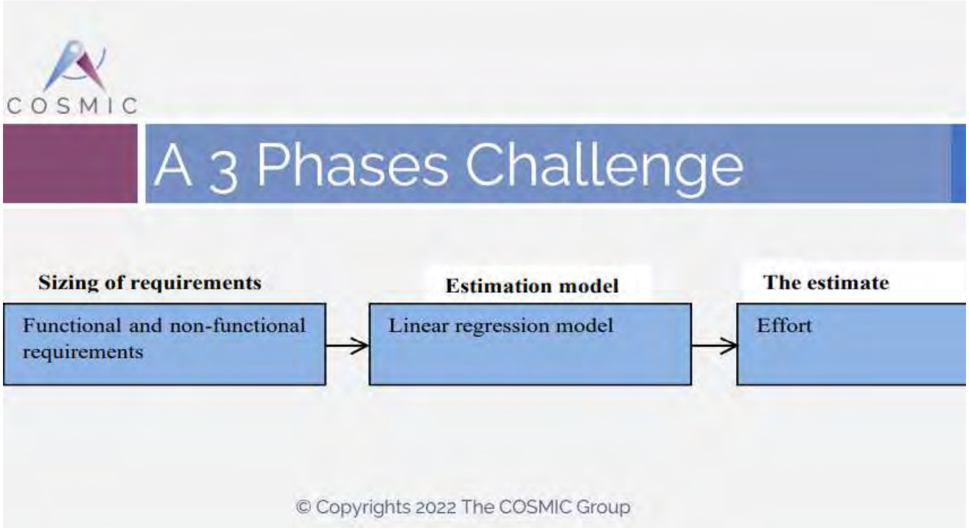
- ✓ 1st prize: 500 Euros
- ✓ 2nd prize: 300 Euros
- ✓ 3rd prize: 200 Euros

Sponsored by the COSMIC Group
cosmic-sizing.org



53 Teams and 234 Students

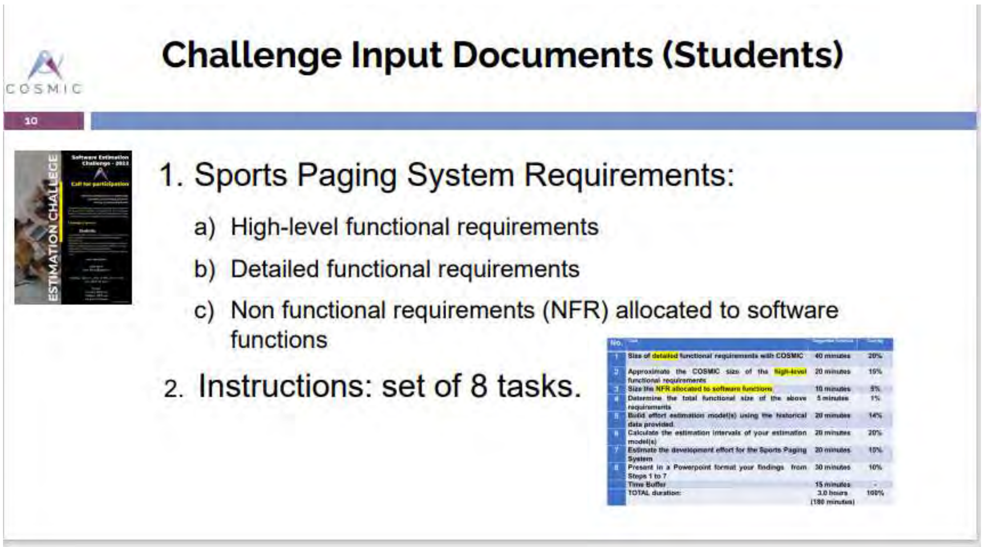
- Atilim University – 21 teams (Turkey)
- Ecole de technologie supérieure – 25 teams (Canada)
- University of Maroua (Cameroon)
- German University in Cairo (Egypt)
- Izmir Institute of Technology (Turkey)
- Universidad National Autonoma de Mexico (Mexico)
- Université du Québec à Montréal – 2 teams (Canada)
- Ecole de technologie supérieure (Canada)



A 3 Phases Challenge

Sizing of requirements: Functional and non-functional requirements → Estimation model: Linear regression model → The estimate: Effort

© Copyrights 2022 The COSMIC Group

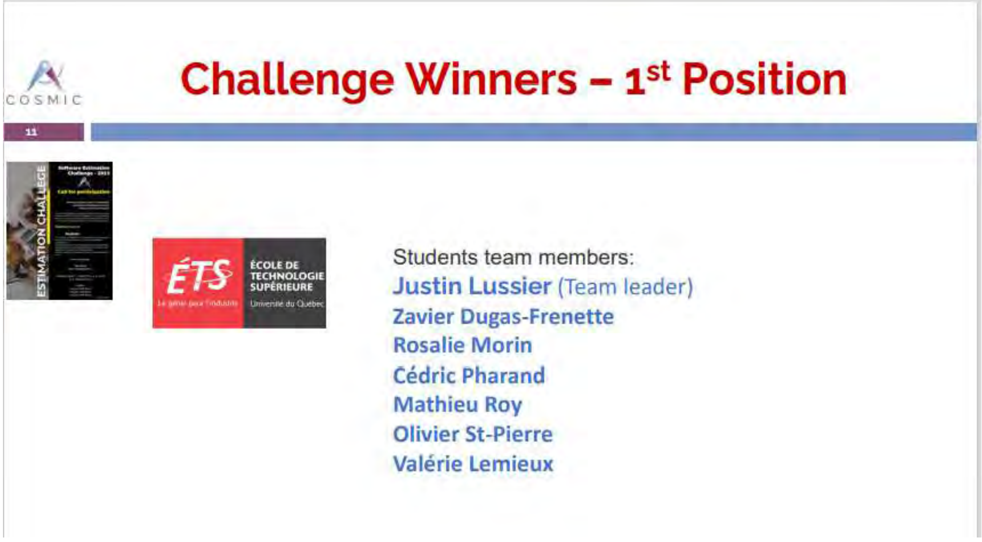


COSMIC
10
ESTIMATION CHALLENGE

Challenge Input Documents (Students)

1. Sports Paging System Requirements:
   a) High-level functional requirements
   b) Detailed functional requirements
   c) Non functional requirements (NFR) allocated to software functions
2. Instructions: set of 8 tasks.

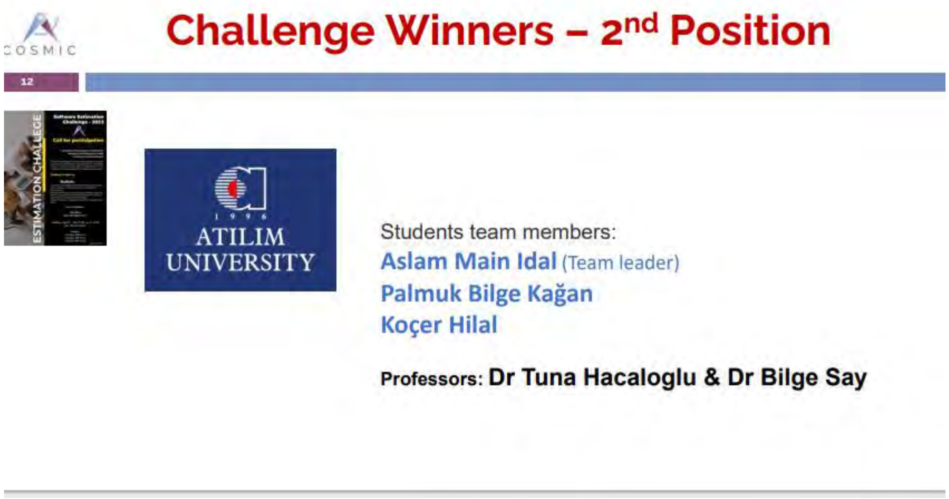
No.	[illegible]	[illegible]	[illegible]
1	Size of detailed functional requirements with COSMIC	40 minutes	20%
2	Approximate the COSMIC size of the high-level functional requirements	20 minutes	10%
3	Size the NFR allocated to software functions	10 minutes	5%
4	Determine the total functional size of the above requirements	5 minutes	1%
5	Build effort estimation model(s) using the historical data provided	20 minutes	14%
6	Calculate the estimation intervals of your estimation model(s)	20 minutes	20%
7	Estimate the development effort for the Sports Paging System	20 minutes	10%
8	Present in a Powerpoint format your findings from Steps 1 to 7	30 minutes	10%
	Time Buffer	15 minutes	-
	TOTAL duration:	3.0 hours (180 minutes)	100%



COSMIC
11
ESTIMATION CHALLENGE
ÉTS
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
Le génie pour l'industrie
Université du Québec

Challenge Winners – 1st Position

Students team members:
Justin Lussier (Team leader)
Zavier Dugas-Frenette
Rosalie Morin
Cédric Pharand
Mathieu Roy
Olivier St-Pierre
Valérie Lemieux

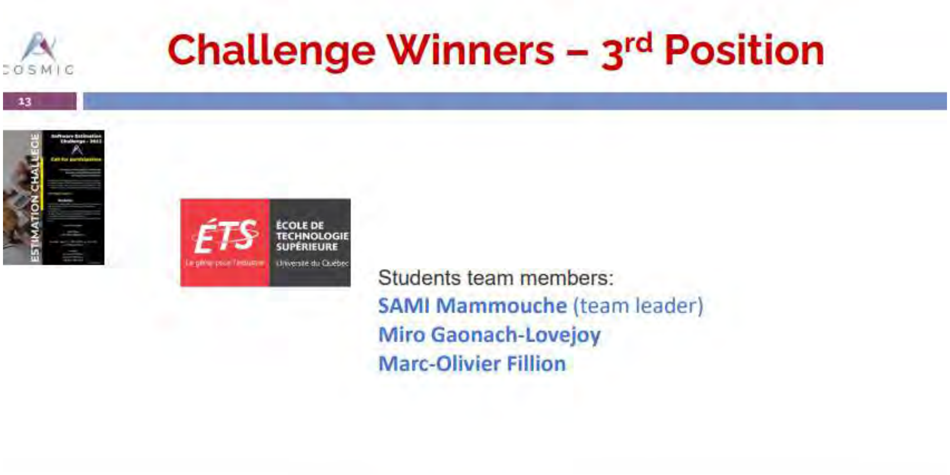


COSMIC
12
ESTIMATION CHALLENGE
1996
ATILIM UNIVERSITY

Challenge Winners – 2nd Position

Students team members:
Aslam Main Idal (Team leader)
Palmuk Bilge Kağan
Koçer Hilal

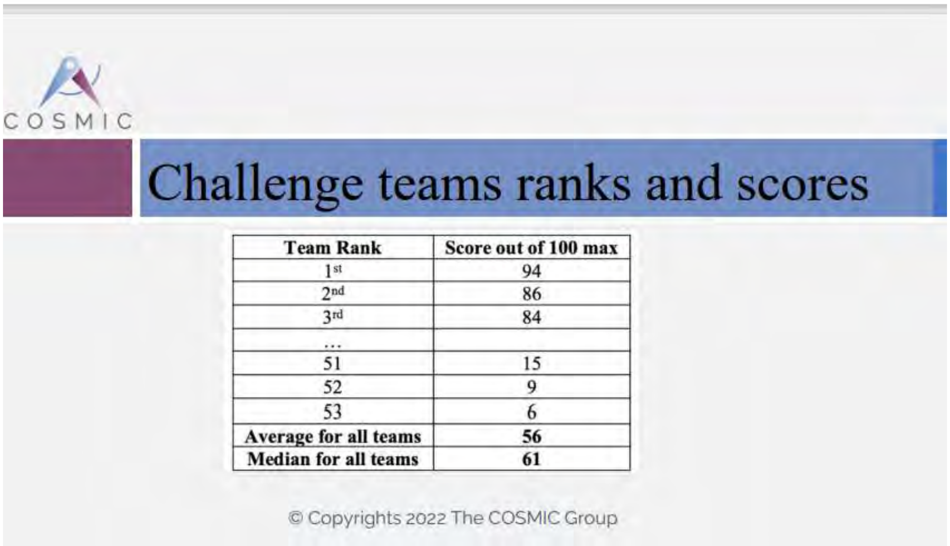
Professors: Dr Tuna Hacaloglu & Dr Bilge Say



COSMIC
13
ESTIMATION CHALLENGE
ÉTS
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
Université du Québec

Challenge Winners – 3rd Position

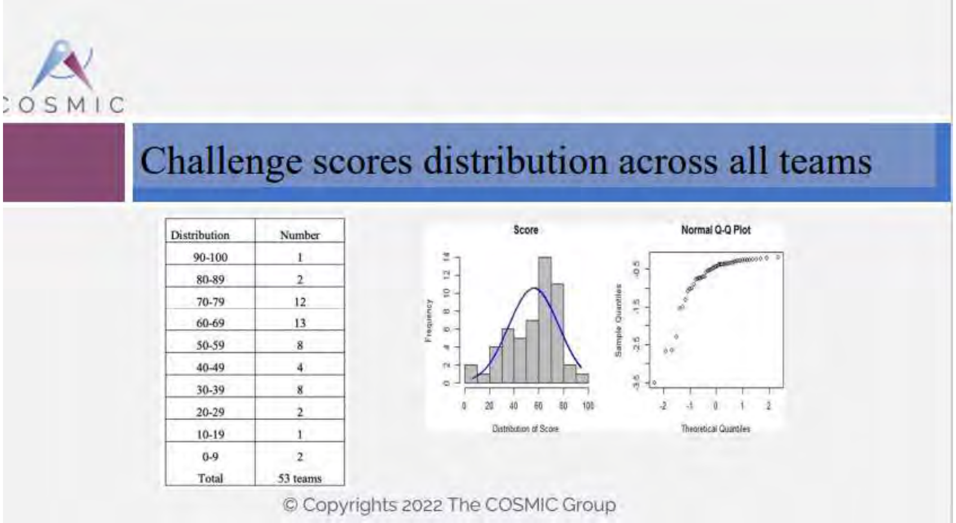
Students team members:
SAMI Mammouche (team leader)
Miro Gaonach-Lovejoy
Marc-Olivier Fillion



Challenge teams ranks and scores

Team Rank	Score out of 100 max
1st	94
2nd	86
3rd	84
…	
51	15
52	9
53	6
Average for all teams	**56**
Median for all teams	**61**

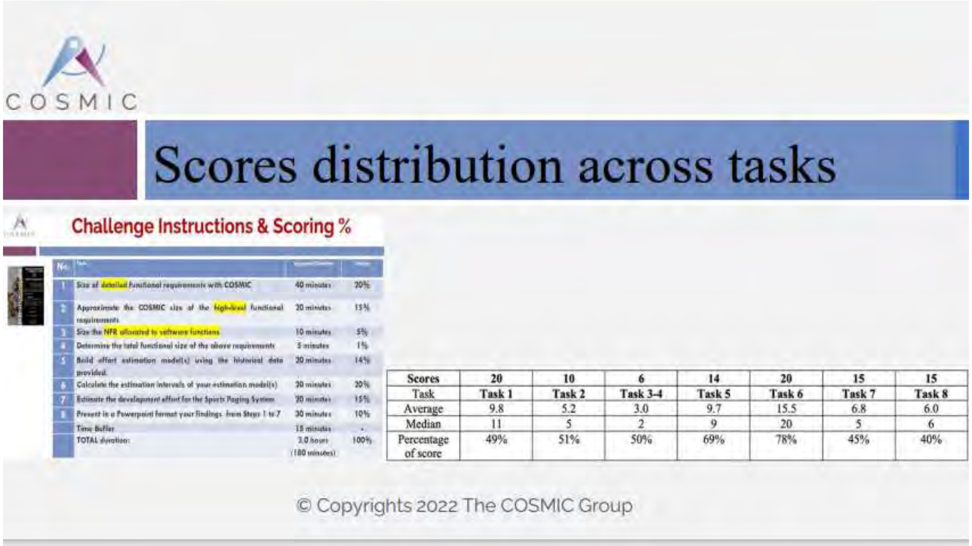
© Copyrights 2022 The COSMIC Group



Challenge scores distribution across all teams

Distribution	Number
90-100	1
80-89	2
70-79	12
60-69	13
50-59	8
40-49	4
30-39	8
20-29	2
10-19	1
0-9	2
Total	53 teams

© Copyrights 2022 The COSMIC Group



COSMIC

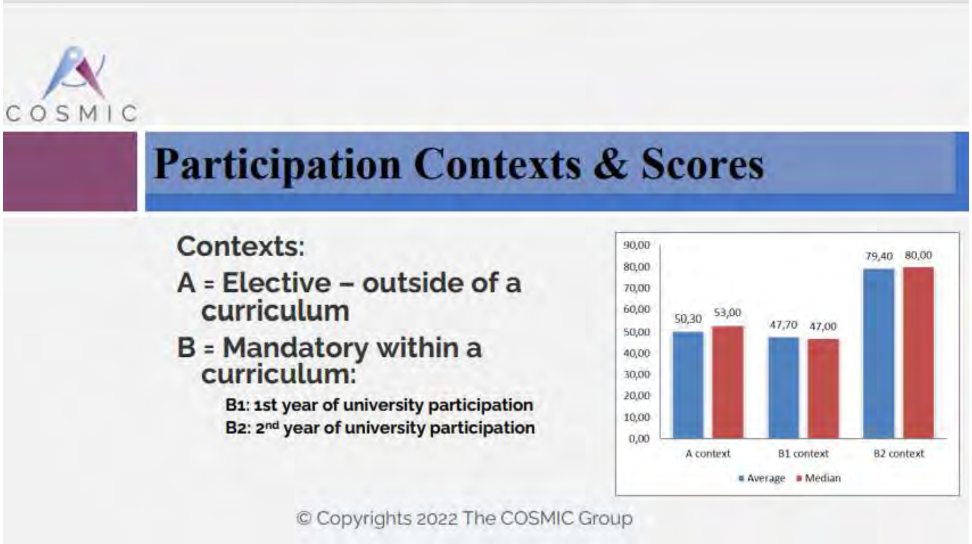
Scores distribution across tasks

Challenge Instructions & Scoring %

No.	[illegible]	[illegible]	[illegible]
1	Size of detailed functional requirements with COSMIC	40 minutes	20%
2	Approximate the COSMIC size of the high-level functional requirements	20 minutes	15%
3	Size the NFR allocated to software functions	10 minutes	5%
4	Determine the total functional size of the above requirements	5 minutes	1%
5	Build effort estimation model(s) using the historical data provided.	20 minutes	14%
6	Calculate the estimation intervals of your estimation model(s)	30 minutes	20%
7	Estimate the development effort for the Sports Paging System	30 minutes	15%
8	Present in a Powerpoint format your findings from Steps 1 to 7	30 minutes	10%
	Time Buffer	15 minutes	-
	TOTAL duration:	3.0 hours (180 minutes)	100%

Scores	20	10	6	14	20	15	15
Task	Task 1	Task 2	Task 3-4	Task 5	Task 6	Task 7	Task 8
Average	9.8	5.2	3.0	9.7	15.5	6.8	6.0
Median	11	5	2	9	20	5	6
Percentage of score	49%	51%	50%	69%	78%	45%	40%

© Copyrights 2022 The COSMIC Group



COSMIC

Participation Contexts & Scores

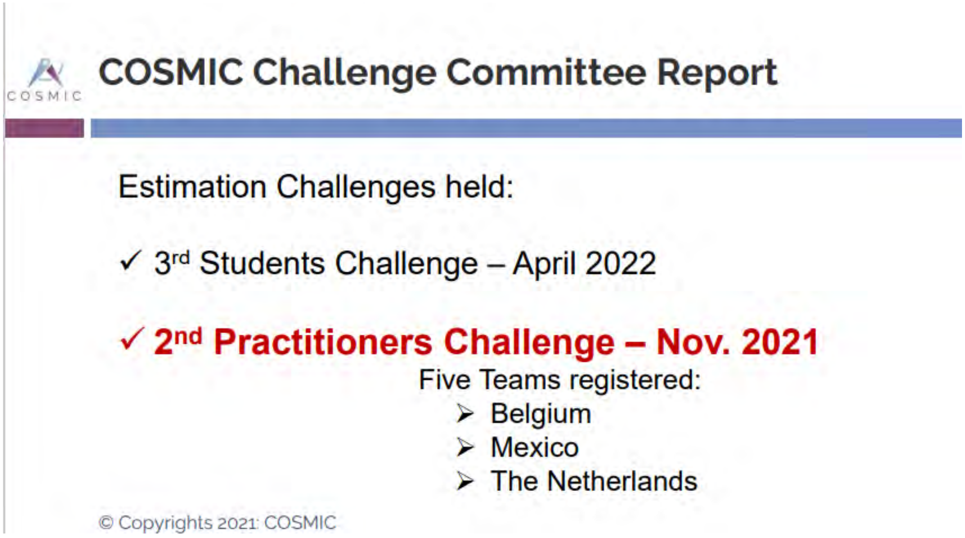
Contexts:

A = Elective – outside of a curriculum

B = Mandatory within a curriculum:

B1: 1st year of university participation
B2: 2nd year of university participation

© Copyrights 2022 The COSMIC Group



COSMIC

COSMIC Challenge Committee Report

Estimation Challenges held:

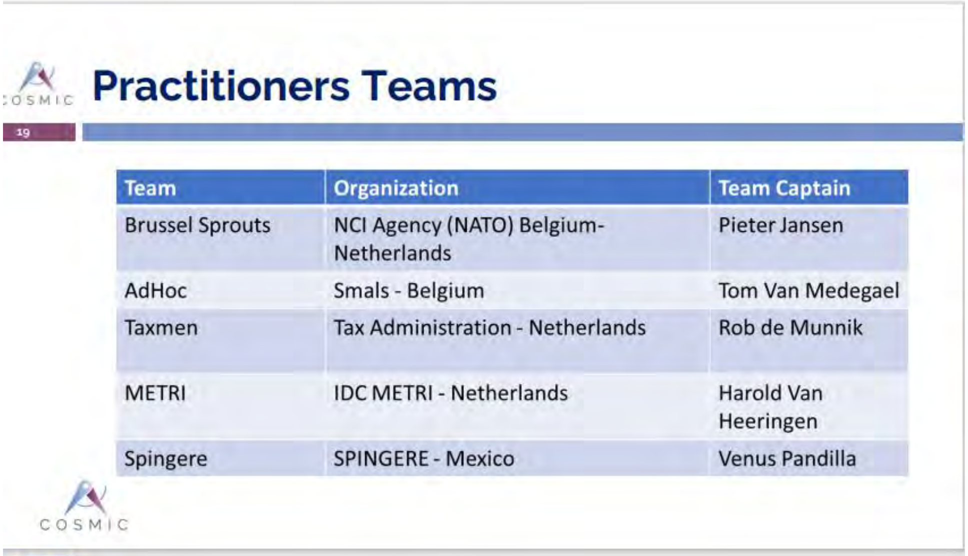
✓ 3rd Students Challenge – April 2022

✓ **2nd Practitioners Challenge – Nov. 2021**

Five Teams registered:

- Belgium
- Mexico
- The Netherlands

© Copyrights 2021: COSMIC



Practitioners Teams

19

Team	Organization	Team Captain
Brussel Sprouts	NCI Agency (NATO) Belgium-Netherlands	Pieter Jansen
AdHoc	Smals - Belgium	Tom Van Medegael
Taxmen	Tax Administration - Netherlands	Rob de Munnik
METRI	IDC METRI - Netherlands	Harold Van Heeringen
Spingere	SPINGERE - Mexico	Venus Pandilla

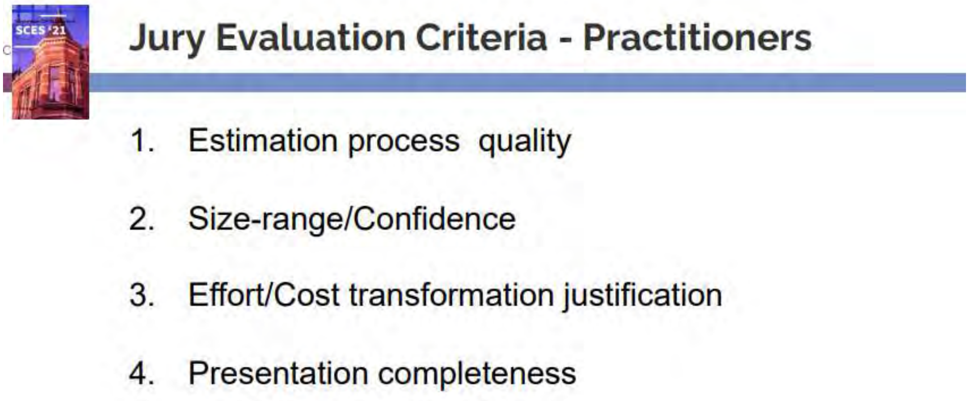
Case Study – Practitioners

20

Marine Research Ship - Seabed Observatory Control System (SOCS)

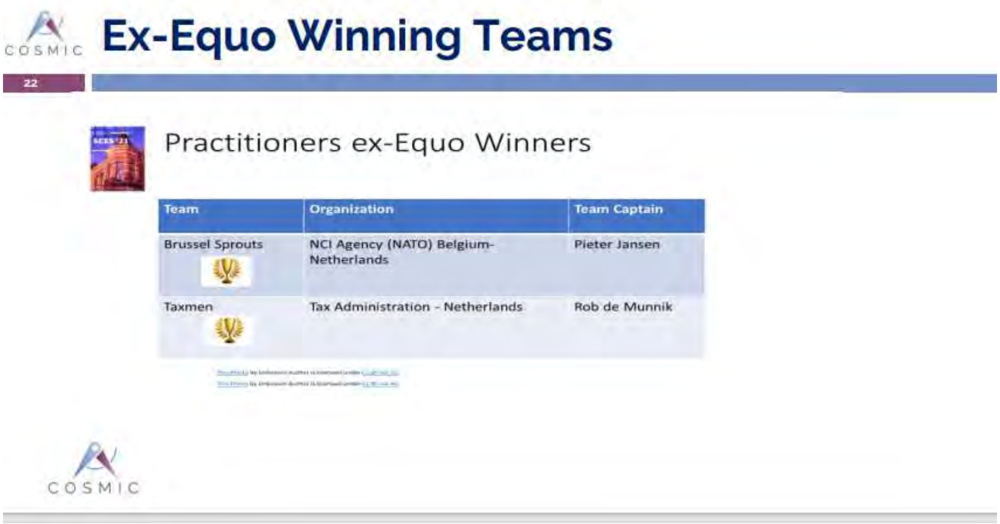
Note: Generic Photos from the web

COSMIC Copyrights 2021 - Software Estimation Challenge 2021



Jury Evaluation Criteria - Practitioners

1. Estimation process quality
2. Size-range/Confidence
3. Effort/Cost transformation justification
4. Presentation completeness



Ex-Equo Winning Teams

22

Practitioners ex-Equo Winners

Team	Organization	Team Captain
Brussel Sprouts 🏆	NCI Agency (NATO) Belgium-Netherlands	Pieter Jansen
Taxmen 🏆	Tax Administration - Netherlands	Rob de Munnik

This Photo by Unknown Author is licensed under [illegible]
This Photo by Unknown Author is licensed under [illegible]

COSMIC

Practitioners Challenge- Thanks to:

Jury Members:

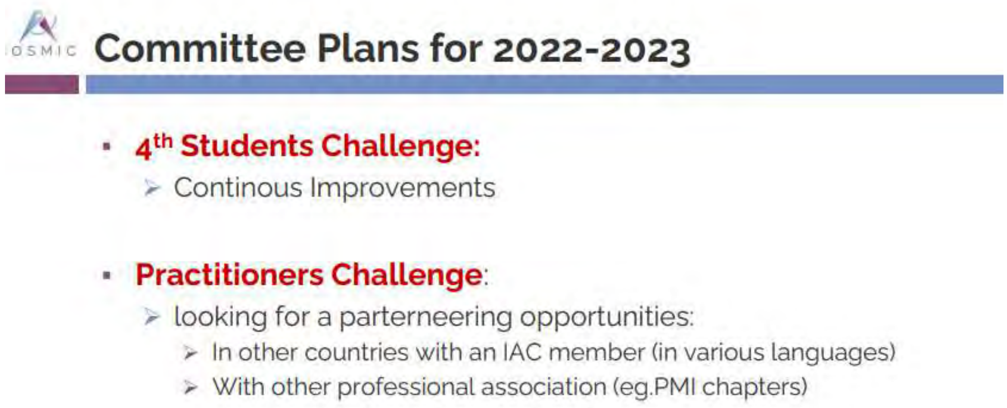
- Thomas Fehlmann – Euro Project Office – Switzerland
- Patrick Hamon – Estimancy – France
- Luca Santillo – Agile Metrics - Italy

Awards presentation:

- Dr. K. Jayakumar – Amitysoft - India

Students Challenge – Thanks to:

All participating teams, Professors-Tutors,
&
Challenge Committee members
(JM Desharnais, D. Koulla Moulla, L. Santillo, A. Abran)



Committee Plans for 2022-2023

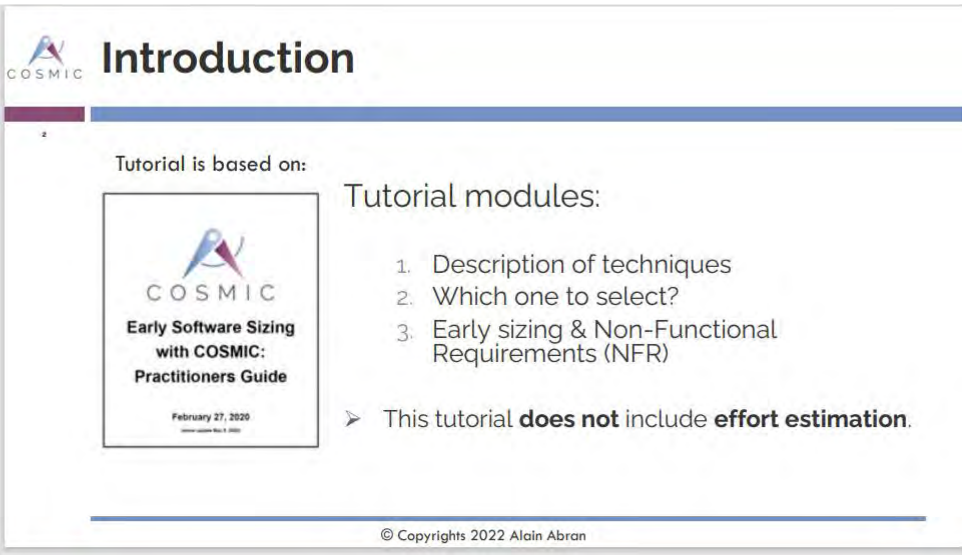
- **4th Students Challenge:**
  - Continous Improvements
- **Practitioners Challenge**:
  - looking for a parterneering opportunities:
    - In other countries with an IAC member (in various languages)
    - With other professional association (eg.PMI chapters)

EARLY SIZING OF REQUIREMENTS FOR ESTIMATION PURPOSES

A Tutorial with COSMIC Sizing – ISO 19761

Alain Abran, ETS Montreal, alain.abran@cosmic-sizing.org

MODULE 1: Description of Techniques



COSMIC

Introduction

2

Tutorial is based on:

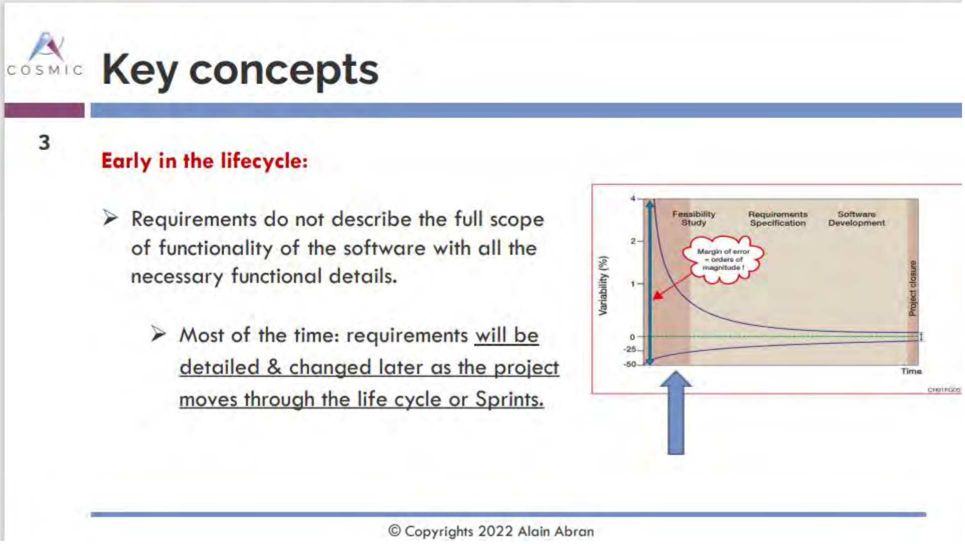
COSMIC
Early Software Sizing with COSMIC: Practitioners Guide
February 27, 2020

Tutorial modules:

1. Description of techniques
2. Which one to select?
3. Early sizing & Non-Functional Requirements (NFR)

➢ This tutorial **does not** include **effort estimation**.

© Copyrights 2022 Alain Abran



COSMIC

Key concepts

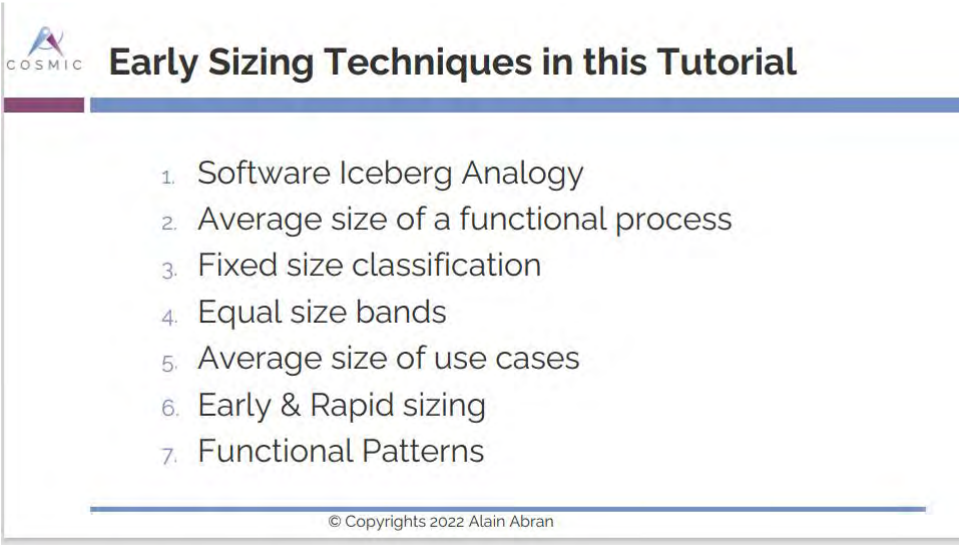
3

Early in the lifecycle:

➢ Requirements do not describe the full scope of functionality of the software with all the necessary functional details.

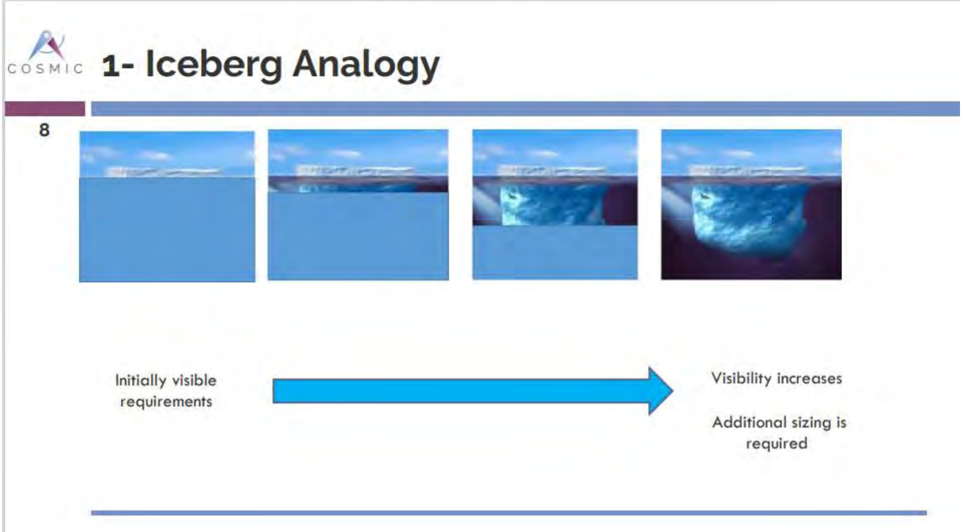
➢ Most of the time: requirements will be detailed & changed later as the project moves through the life cycle or Sprints.

© Copyrights 2022 Alain Abran



Early Sizing Techniques in this Tutorial

1. Software Iceberg Analogy
2. Average size of a functional process
3. Fixed size classification
4. Equal size bands
5. Average size of use cases
6. Early & Rapid sizing
7. Functional Patterns

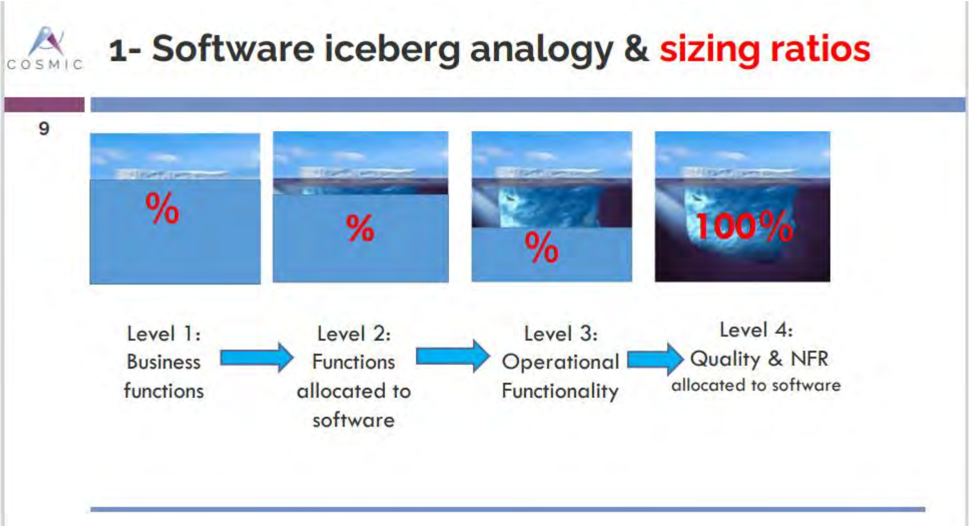


1- Iceberg Analogy

8

Initially visible requirements → Visibility increases

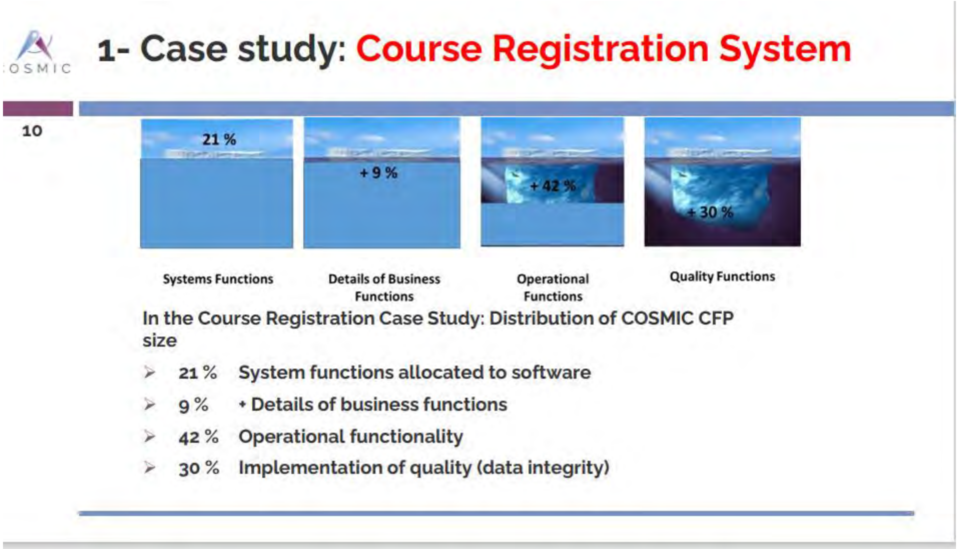
Additional sizing is required



1- Software iceberg analogy & sizing ratios

9

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



1- Case study: Course Registration System

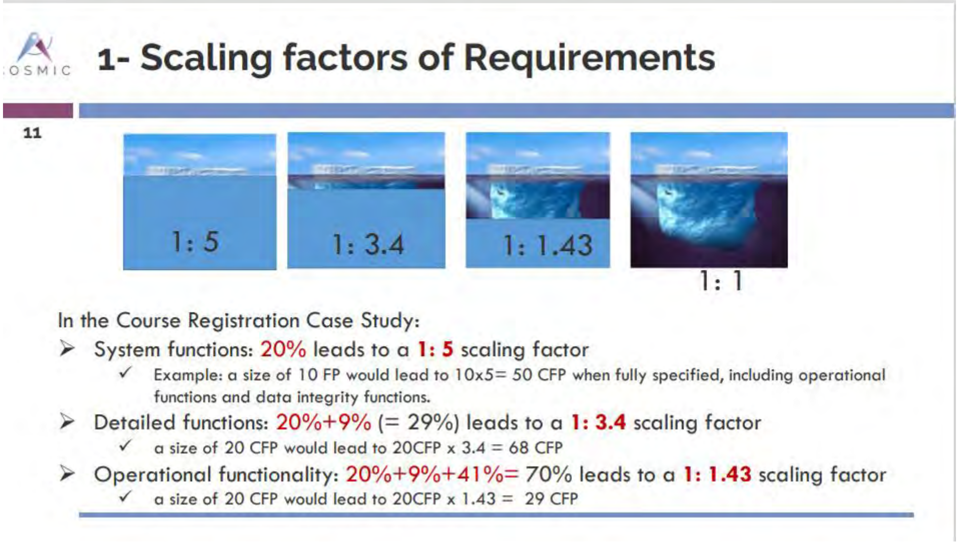
10

21 % | + 9 % | + 42 % | + 30 %

Systems Functions | Details of Business Functions | Operational Functions | Quality Functions

In the Course Registration Case Study: Distribution of COSMIC CFP size

- 21 % System functions allocated to software
- 9 % + Details of business functions
- 42 % Operational functionality
- 30 % Implementation of quality (data integrity)



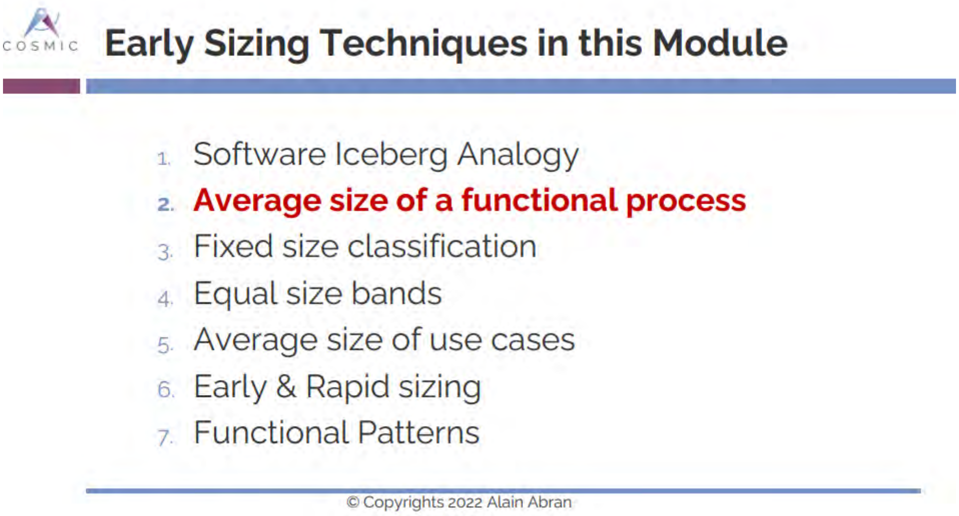
1- Scaling factors of Requirements

11

1: 5 | 1: 3.4 | 1: 1.43 | 1: 1

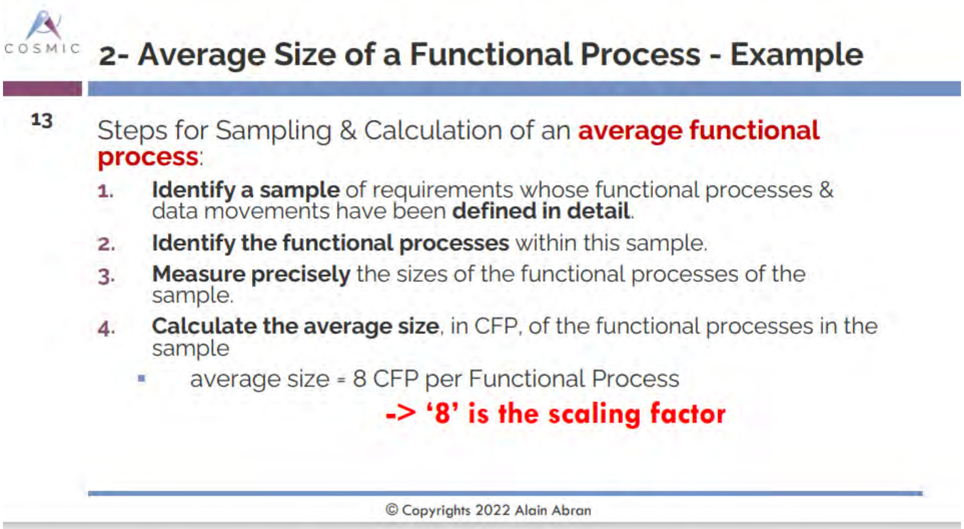
In the Course Registration Case Study:

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



Early Sizing Techniques in this Module

1. Software Iceberg Analogy
2. **Average size of a functional process**
3. Fixed size classification
4. Equal size bands
5. Average size of use cases
6. Early & Rapid sizing
7. Functional Patterns



2- Average Size of a Functional Process - Example

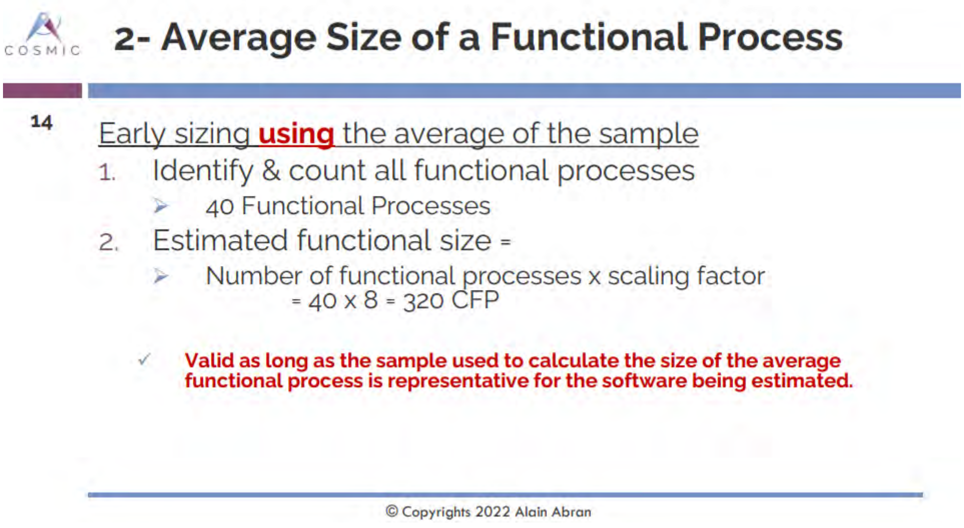
13

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

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

-> '8' is the scaling factor

© Copyrights 2022 Alain Abran



2- Average Size of a Functional Process

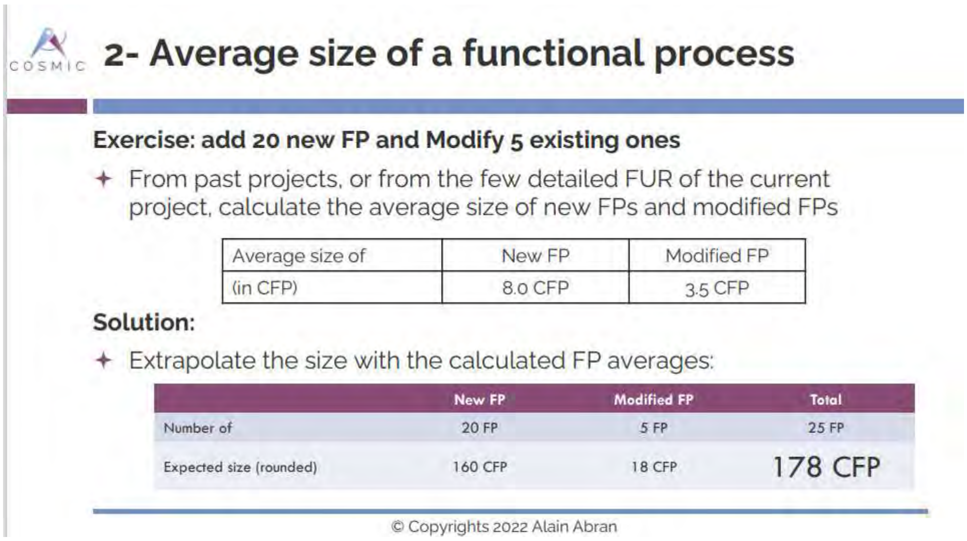
14

Early sizing **using** the average of the sample

1. Identify & count all functional processes
   - 40 Functional Processes
2. Estimated functional size =
   - Number of functional processes x scaling factor
     = 40 x 8 = 320 CFP

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

© Copyrights 2022 Alain Abran



2- Average size of a functional process

Exercise: add 20 new FP and Modify 5 existing ones

- From past projects, or from the few detailed FUR of the current project, calculate the average size of new FPs and modified FPs

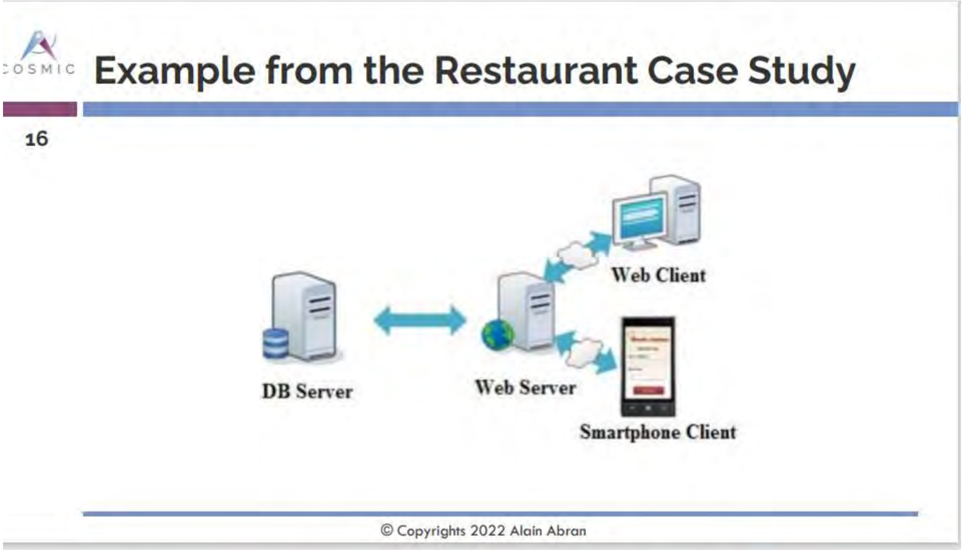
Average size of	New FP	Modified FP
(in CFP)	8.0 CFP	3.5 CFP

Solution:

- Extrapolate the size with the calculated FP averages:

	New FP	Modified FP	Total
Number of	20 FP	5 FP	25 FP
Expected size (rounded)	160 CFP	18 CFP	178 CFP

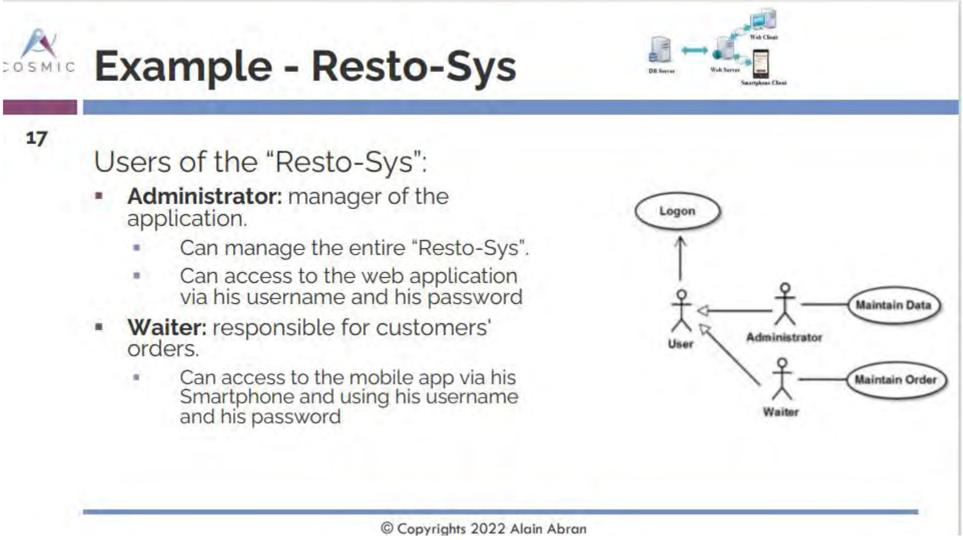
© Copyrights 2022 Alain Abran



COSMIC

Example from the Restaurant Case Study

16



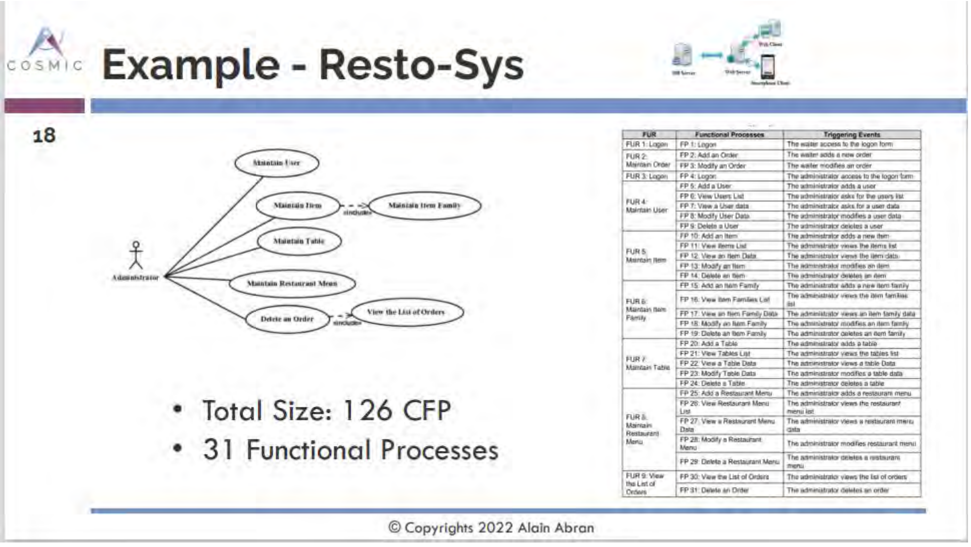
COSMIC

Example - Resto-Sys

17

Users of the "Resto-Sys":

- **Administrator:** manager of the application.
  - Can manage the entire "Resto-Sys".
  - Can access to the web application via his username and his password
- **Waiter:** responsible for customers' orders.
  - Can access to the mobile app via his Smartphone and using his username and his password



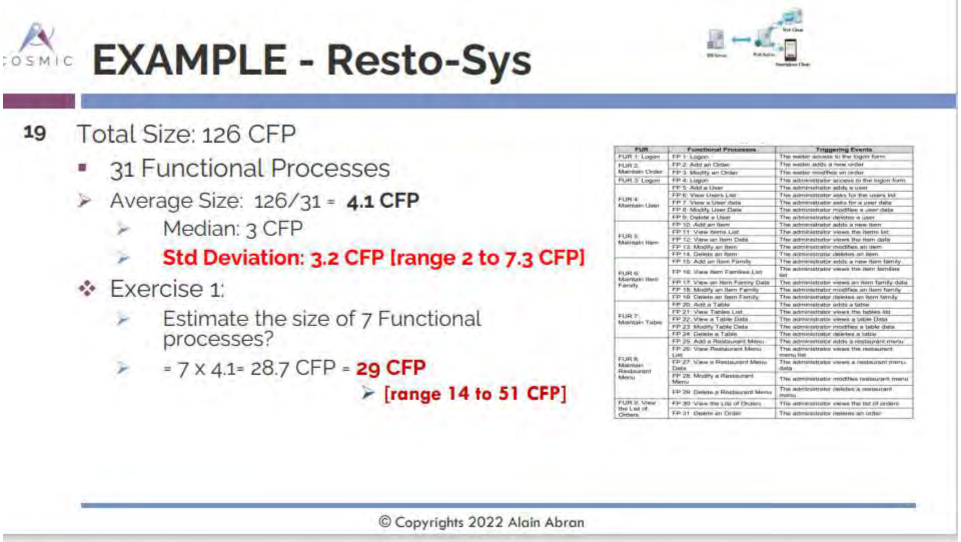
COSMIC

Example - Resto-Sys

18

- Total Size: 126 CFP
- 31 Functional Processes

FUR	Functional Processes	Triggering Events
FUR 1: Logon	FP 1: Logon	The waiter access to the logon form
FUR 2: Maintain Order	FP 2: Add an Order	The waiter adds a new order
	FP 3: Modify an Order	The waiter modifies an order
FUR 3: Logon	FP 4: Logon	The administrator access to the logon form
FUR 4: Maintain User	FP 5: Add a User	The administrator adds a user
	FP 6: View Users List	The administrator asks for the users list
	FP 7: View a User data	The administrator asks for a user data
	FP 8: Modify User Data	The administrator modifies a user data
	FP 9: Delete a User	The administrator deletes a user
FUR 5: Maintain Item	FP 10: Add an Item	The administrator adds a new item
	FP 11: View items List	The administrator views the items list
	FP 12: View an Item Data	The administrator views the item data
	FP 13: Modify an Item	The administrator modifies an item
	FP 14: Delete an Item	The administrator deletes an item
FUR 6: Maintain Item Family	FP 15: Add an Item Family	The administrator adds a new item family
	FP 16: View Item Families List	The administrator views the item families list
	FP 17: View an Item Family Data	The administrator views an item family data
	FP 18: Modify an Item Family	The administrator modifies an item family
	FP 19: Delete an Item Family	The administrator deletes an item family
FUR 7: Maintain Table	FP 20: Add a Table	The administrator adds a table
	FP 21: View Tables List	The administrator views the tables list
	FP 22: View a Table Data	The administrator views a table Data
	FP 23: Modify Table Data	The administrator modifies a table data
	FP 24: Delete a Table	The administrator deletes a table
FUR 8: Maintain Restaurant Menu	FP 25: Add a Restaurant Menu	The administrator adds a restaurant menu
	FP 26: View Restaurant Menu List	The administrator views the restaurant menu list
	FP 27: View a Restaurant Menu Data	The administrator views a restaurant menu data
	FP 28: Modify a Restaurant Menu	The administrator modifies restaurant menu
	FP 29: Delete a Restaurant Menu	The administrator deletes a restaurant menu
FUR 9: View the List of Orders	FP 30: View the List of Orders	The administrator views the list of orders
	FP 31: Delete an Order	The administrator deletes an order



EXAMPLE - Resto-Sys

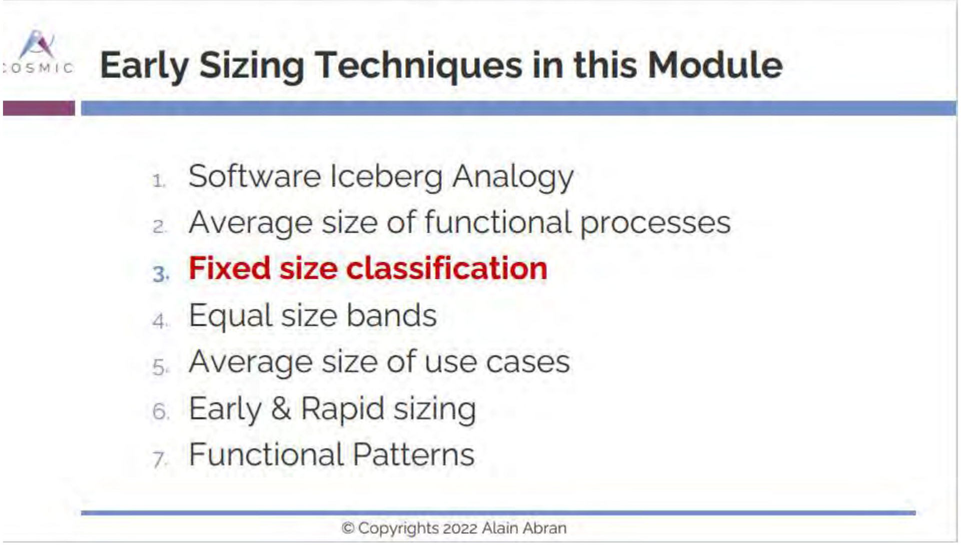
19

Total Size: 126 CFP

- 31 Functional Processes
- Average Size: 126/31 = **4.1 CFP**
  - Median: 3 CFP
  - **Std Deviation: 3.2 CFP [range 2 to 7.3 CFP]**
- Exercise 1:
  - Estimate the size of 7 Functional processes?
  - = 7 x 4.1= 28.7 CFP = **29 CFP**
    - **[range 14 to 51 CFP]**

FUR	Functional Processes	Triggering Events
FUR 1: Logon	FP 1: Logon	The waiter access to the logon form
FUR 2: Maintain Order	FP 2: Add an Order	The waiter adds a new order
	FP 3: Modify an Order	The waiter modifies an order
FUR 3: Logon	FP 4: Logon	The administrator access to the logon form
FUR 4: Maintain User	FP 5: Add a User	The administrator adds a user
	FP 6: View Users List	The administrator asks for the users list
	FP 7: View a User data	The administrator asks for a user data
	FP 8: Modify User Data	The administrator modifies a user data
	FP 9: Delete a User	The administrator deletes a user
FUR 5: Maintain Item	FP 10: Add an Item	The administrator adds a new item
	FP 11: View Items List	The administrator views the items list
	FP 12: View an Item Data	The administrator views the item data
	FP 13: Modify an Item	The administrator modifies an item
	FP 14: Delete an Item	The administrator deletes an item
FUR 6: Maintain Item Family	FP 15: Add an Item Family	The administrator adds a new item family
	FP 16: View Item Families List	The administrator views the item families list
	FP 17: View an Item Family Data	The administrator views an item family data
	FP 18: Modify an Item Family	The administrator modifies an item family
	FP 19: Delete an Item Family	The administrator deletes an item family
FUR 7: Maintain Table	FP 20: Add a Table	The administrator adds a table
	FP 21: View Tables List	The administrator views the tables list
	FP 22: View a Table Data	The administrator views a table Data
	FP 23: Modify Table Data	The administrator modifies a table data
	FP 24: Delete a Table	The administrator deletes a table
FUR 8: Maintain Restaurant Menu	FP 25: Add a Restaurant Menu	The administrator adds a restaurant menu
	FP 26: View Restaurant Menu List	The administrator views the restaurant menu list
	FP 27: View a Restaurant Menu Data	The administrator views a restaurant menu data
	FP 28: Modify a Restaurant Menu	The administrator modifies restaurant menu
	FP 29: Delete a Restaurant Menu	The administrator deletes a restaurant menu
FUR 9: View the List of Orders	FP 30: View the List of Orders	The administrator views the list of orders
	FP 31: Delete an Order	The administrator deletes an order

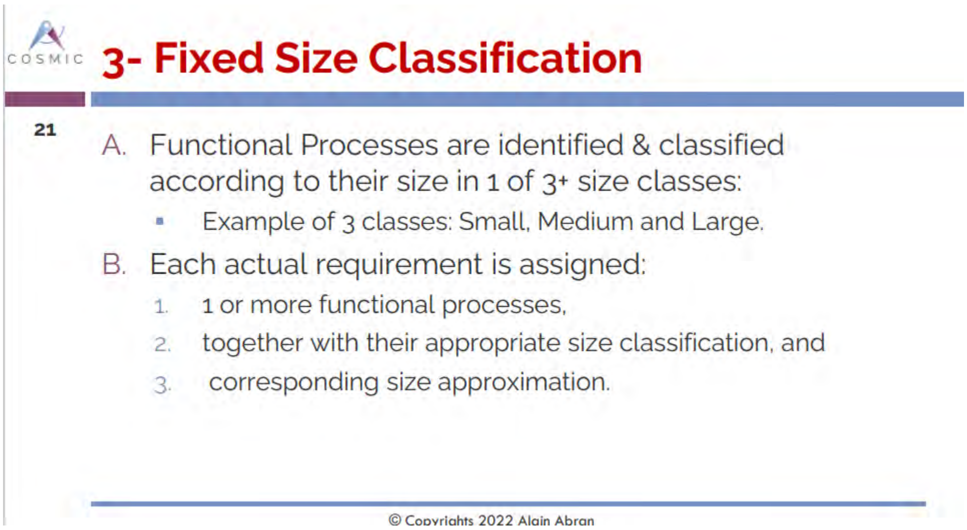
© Copyrights 2022 Alain Abran



Early Sizing Techniques in this Module

1. Software Iceberg Analogy
2. Average size of functional processes
3. **Fixed size classification**
4. Equal size bands
5. Average size of use cases
6. Early & Rapid sizing
7. Functional Patterns

© Copyrights 2022 Alain Abran



3- Fixed Size Classification

21

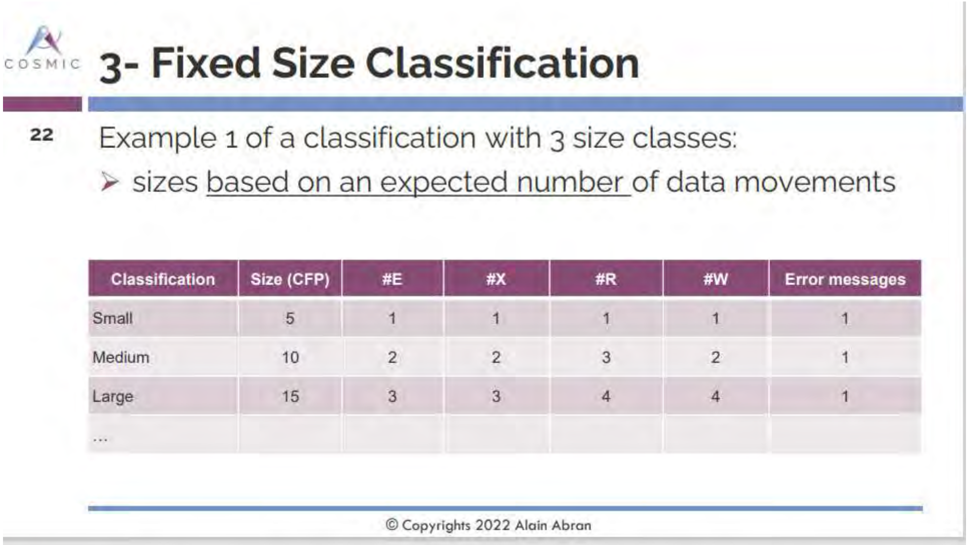
A. Functional Processes are identified & classified according to their size in 1 of 3+ size classes:

- Example of 3 classes: Small, Medium and Large.

B. Each actual requirement is assigned:

1. 1 or more functional processes,
2. together with their appropriate size classification, and
3. corresponding size approximation.

© Copyrights 2022 Alain Abran

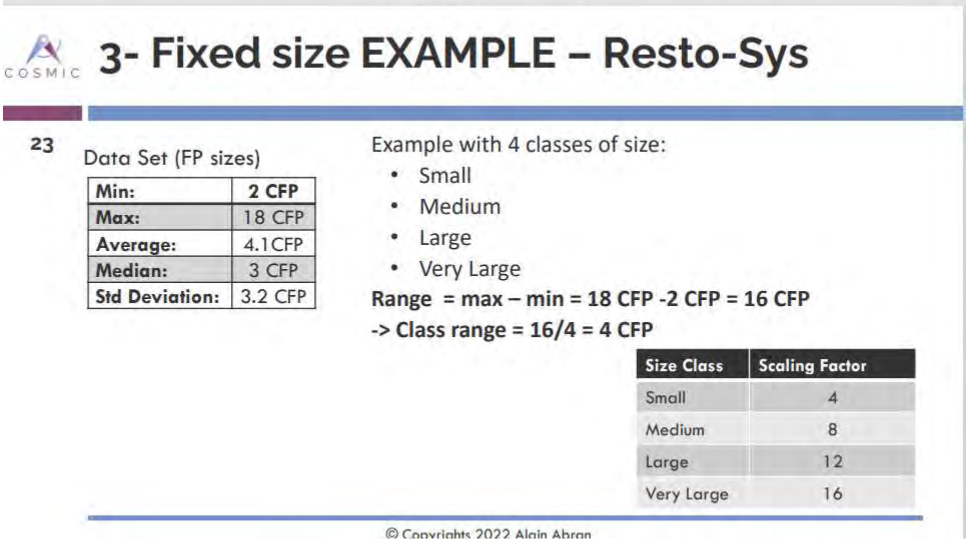


3- Fixed Size Classification

22 Example 1 of a classification with 3 size classes:

- sizes based on an expected number of data movements

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



3- Fixed size EXAMPLE – Resto-Sys

23 Data Set (FP sizes)

Min:	2 CFP
Max:	18 CFP
Average:	4.1 CFP
Median:	3 CFP
Std Deviation:	3.2 CFP

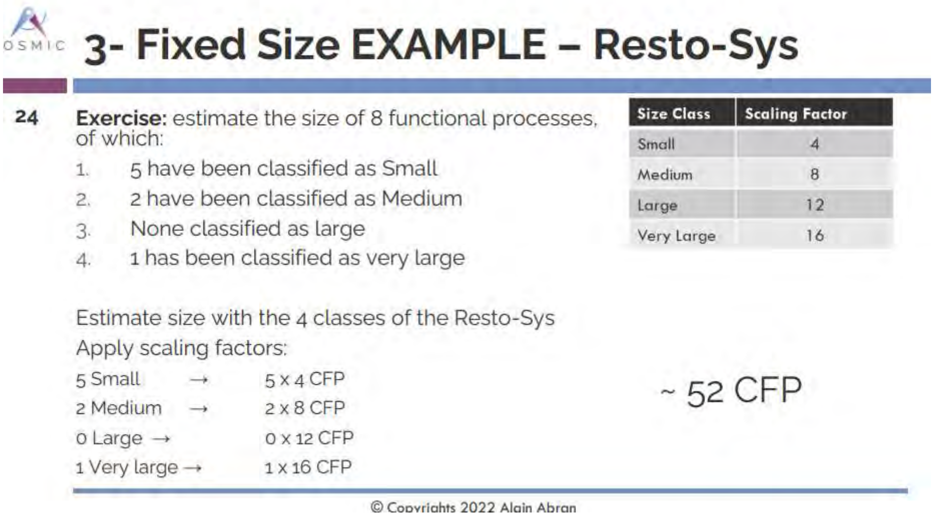
Example with 4 classes of size:

- Small
- Medium
- Large
- Very Large

Range = max – min = 18 CFP -2 CFP = 16 CFP

-> Class range = 16/4 = 4 CFP

Size Class	Scaling Factor
Small	4
Medium	8
Large	12
Very Large	16



3- Fixed Size EXAMPLE – Resto-Sys

24 **Exercise:** estimate the size of 8 functional processes, of which:

1. 5 have been classified as Small
2. 2 have been classified as Medium
3. None classified as large
4. 1 has been classified as very large

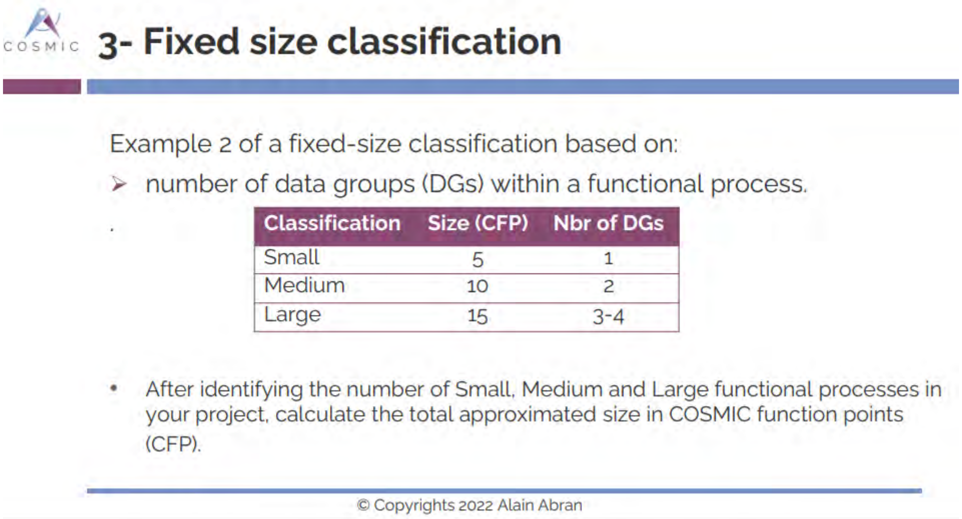
Size Class	Scaling Factor
Small	4
Medium	8
Large	12
Very Large	16

Estimate size with the 4 classes of the Resto-Sys

Apply scaling factors:

5 Small →	5 x 4 CFP
2 Medium →	2 x 8 CFP
0 Large →	0 x 12 CFP
1 Very large →	1 x 16 CFP

~ 52 CFP



3- Fixed size classification

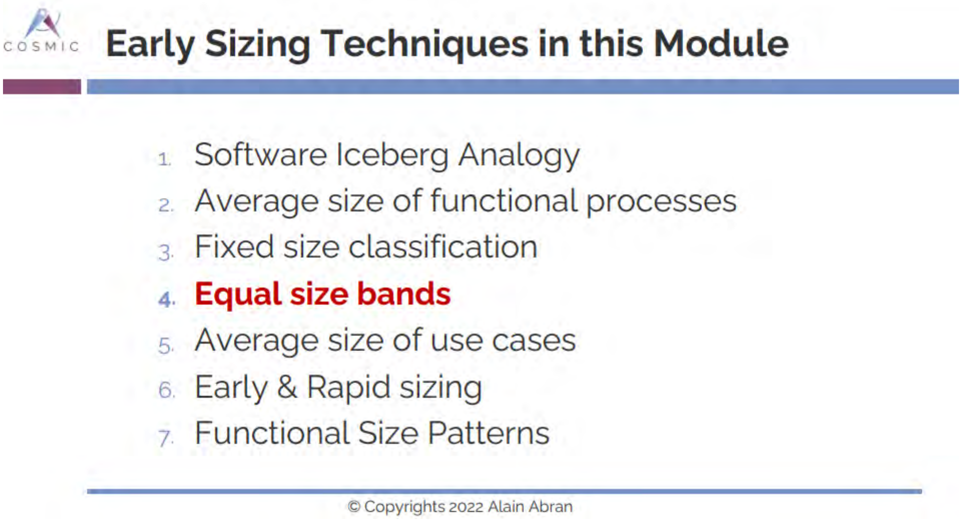
Example 2 of a fixed-size classification based on:

- number of data groups (DGs) within a functional process.

Classification	Size (CFP)	Nbr of DGs
Small	5	1
Medium	10	2
Large	15	3-4

- After identifying the number of Small, Medium and Large functional processes in your project, calculate the total approximated size in COSMIC function points (CFP).

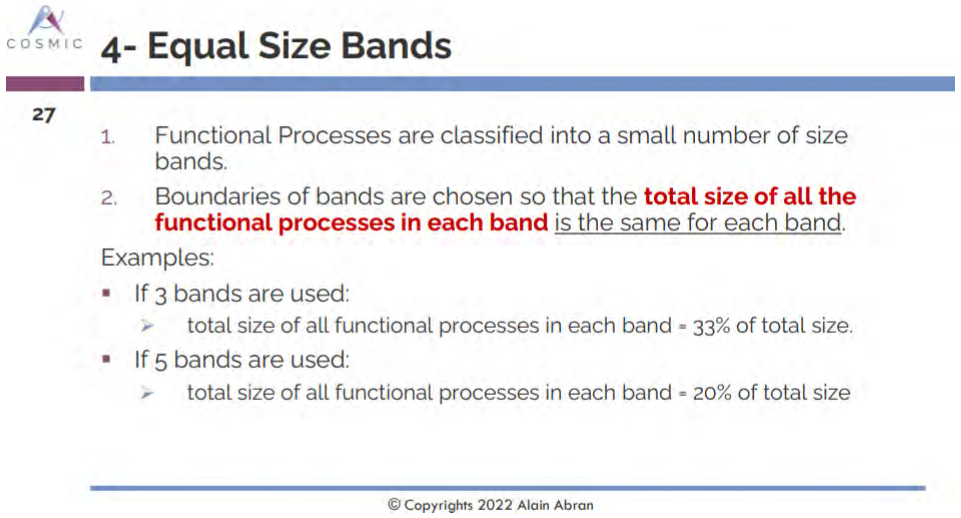
© Copyrights 2022 Alain Abran



Early Sizing Techniques in this Module

1. Software Iceberg Analogy
2. Average size of functional processes
3. Fixed size classification
4. **Equal size bands**
5. Average size of use cases
6. Early & Rapid sizing
7. Functional Size Patterns

© Copyrights 2022 Alain Abran



4- Equal Size Bands

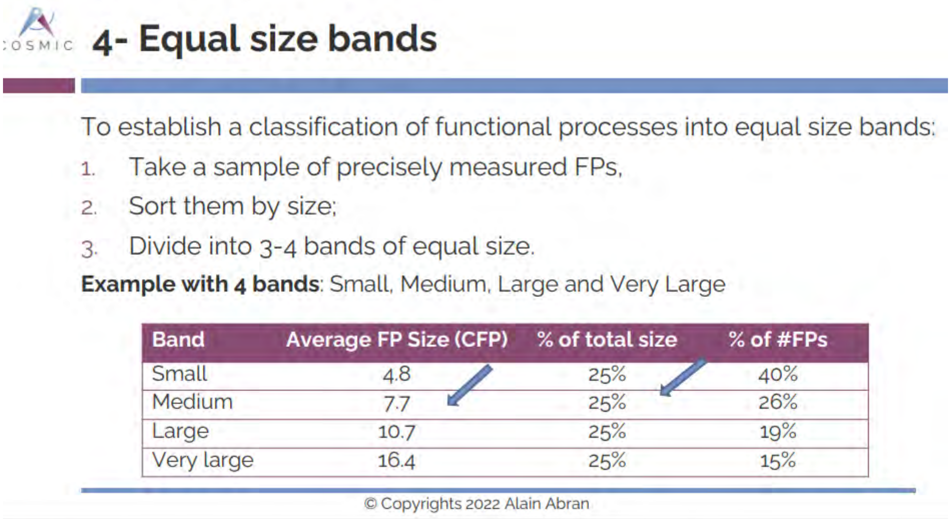
27

1. Functional Processes are classified into a small number of size bands.
2. Boundaries of bands are chosen so that the **total size of all the functional processes in each band** is the same for each band.

Examples:

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

© Copyrights 2022 Alain Abran



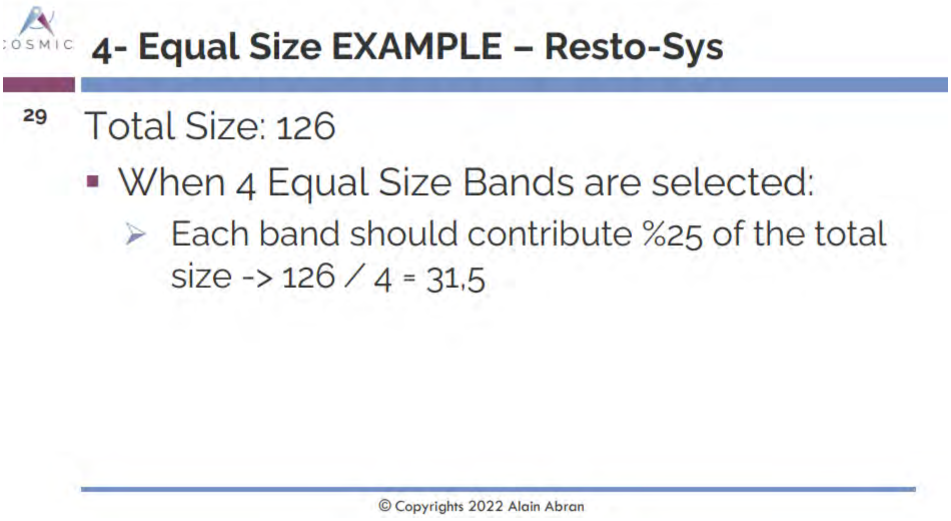
4- Equal size bands

To establish a classification of functional processes into equal size bands:

1. Take a sample of precisely measured FPs,
2. Sort them by size;
3. Divide into 3-4 bands of equal size.

Example with 4 bands: Small, Medium, Large and Very Large

Band	Average FP Size (CFP)	% of total size	% of #FPs
Small	4.8	25%	40%
Medium	7.7	25%	26%
Large	10.7	25%	19%
Very large	16.4	25%	15%

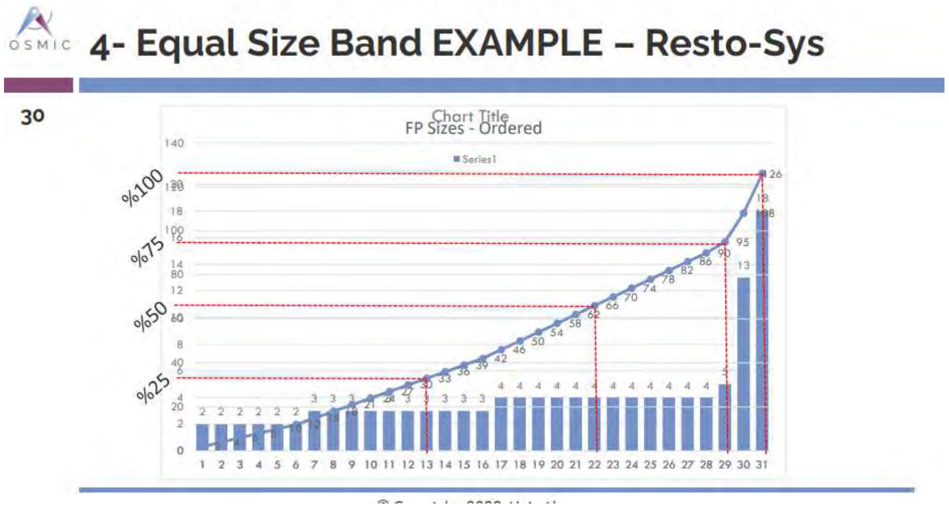


4- Equal Size EXAMPLE – Resto-Sys

29

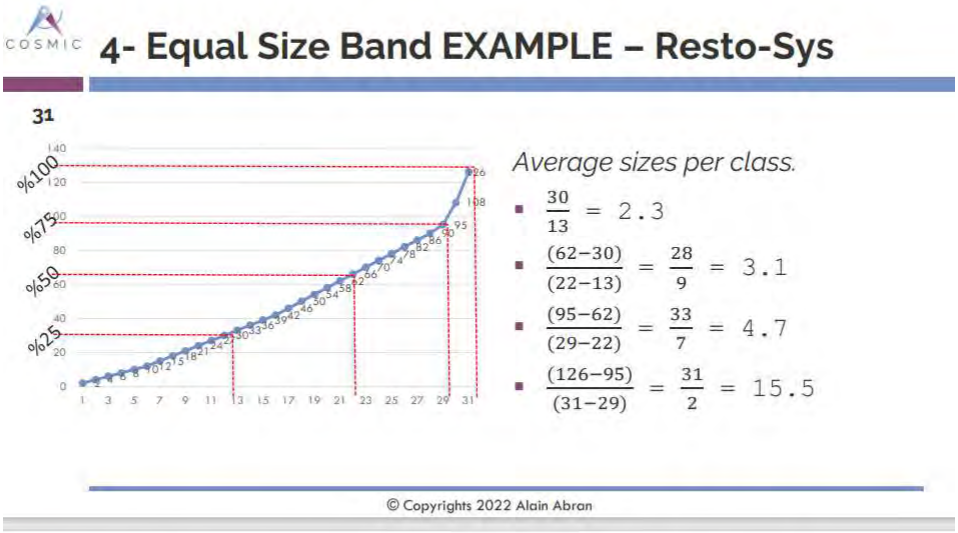
Total Size: 126

- When 4 Equal Size Bands are selected:
  - Each band should contribute %25 of the total size -> 126 / 4 = 31,5



4- Equal Size Band EXAMPLE – Resto-Sys

30

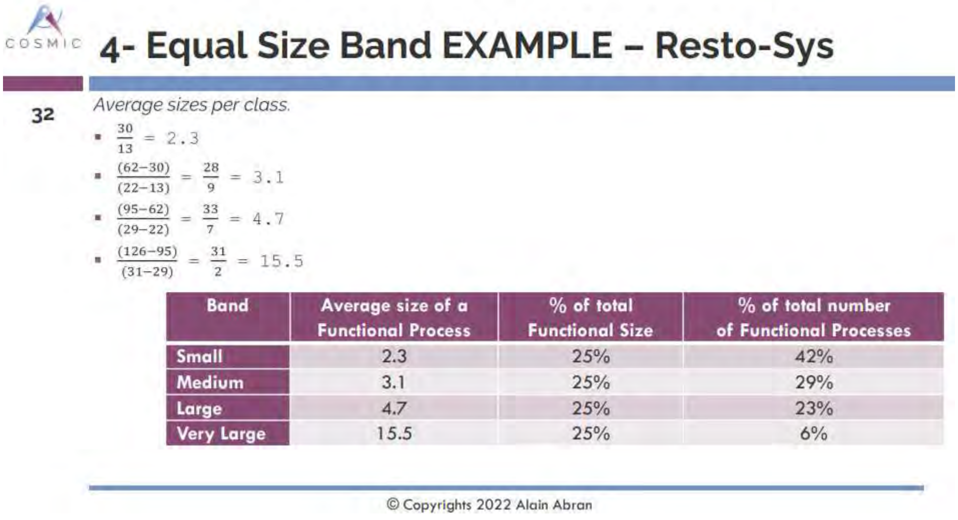


4- Equal Size Band EXAMPLE – Resto-Sys

31

Average sizes per class.

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



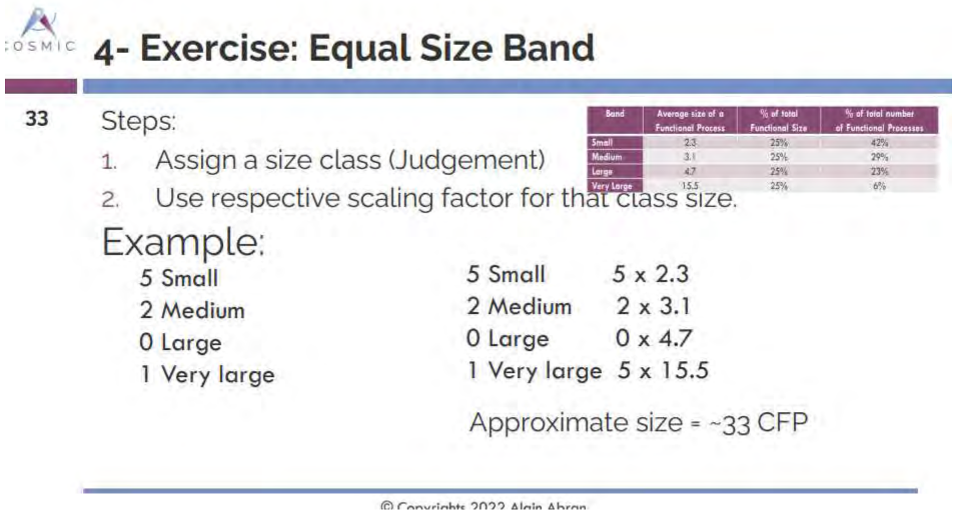
4- Equal Size Band EXAMPLE – Resto-Sys

32

Average sizes per class.

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

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



4- Exercise: Equal Size Band

33

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

Steps:

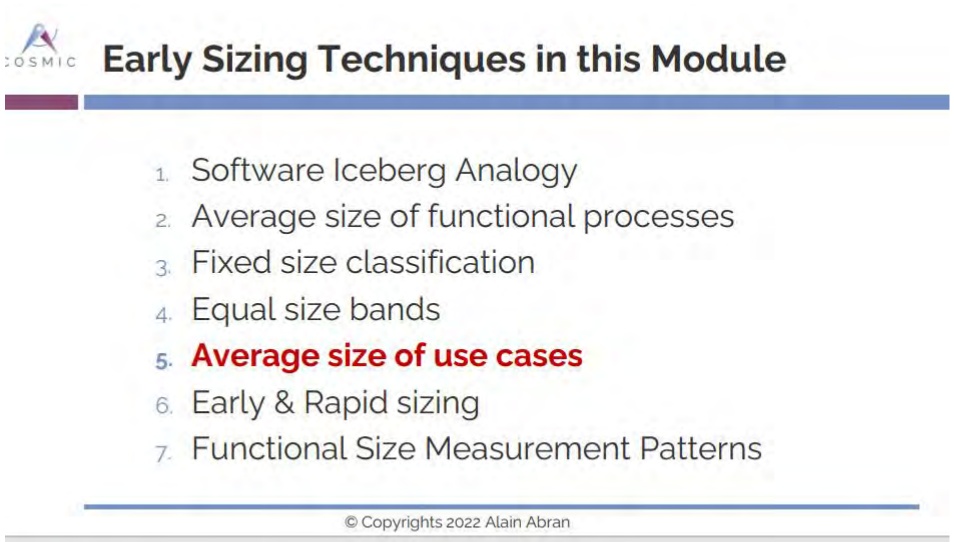
1. Assign a size class (Judgement)
2. Use respective scaling factor for that class size.

Example:

5 Small
2 Medium
0 Large
1 Very large

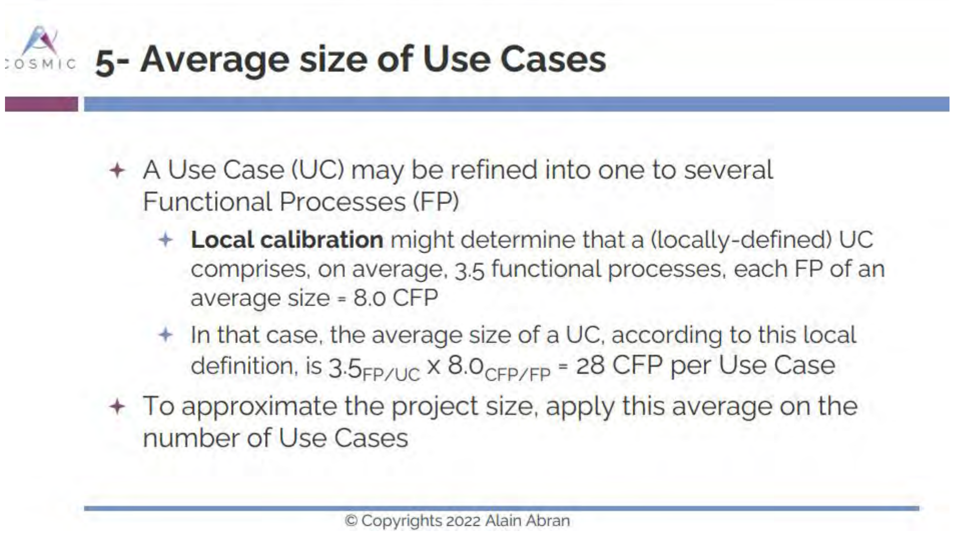
5 Small	5 x 2.3
2 Medium	2 x 3.1
0 Large	0 x 4.7
1 Very large	5 x 15.5

Approximate size = ~33 CFP



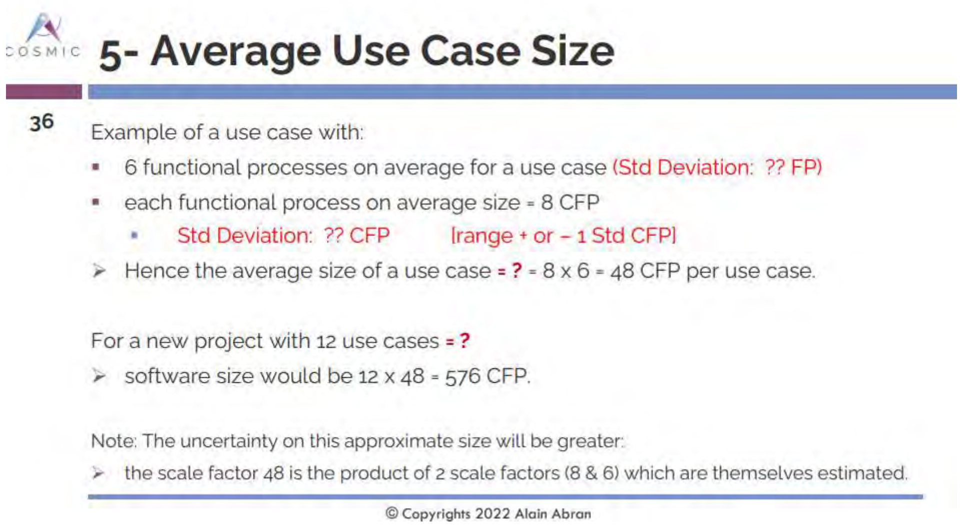
Early Sizing Techniques in this Module

1. Software Iceberg Analogy
2. Average size of functional processes
3. Fixed size classification
4. Equal size bands
5. **Average size of use cases**
6. Early & Rapid sizing
7. Functional Size Measurement Patterns



5- Average size of Use Cases

- A Use Case (UC) may be refined into one to several Functional Processes (FP)
  - **Local calibration** might determine that a (locally-defined) UC comprises, on average, 3.5 functional processes, each FP of an average size = 8.0 CFP
  - In that case, the average size of a UC, according to this local definition, is $3.5_{FP/UC}$ x $8.0_{CFP/FP}$ = 28 CFP per Use Case
- To approximate the project size, apply this average on the number of Use Cases



5- Average Use Case Size

36

Example of a use case with:

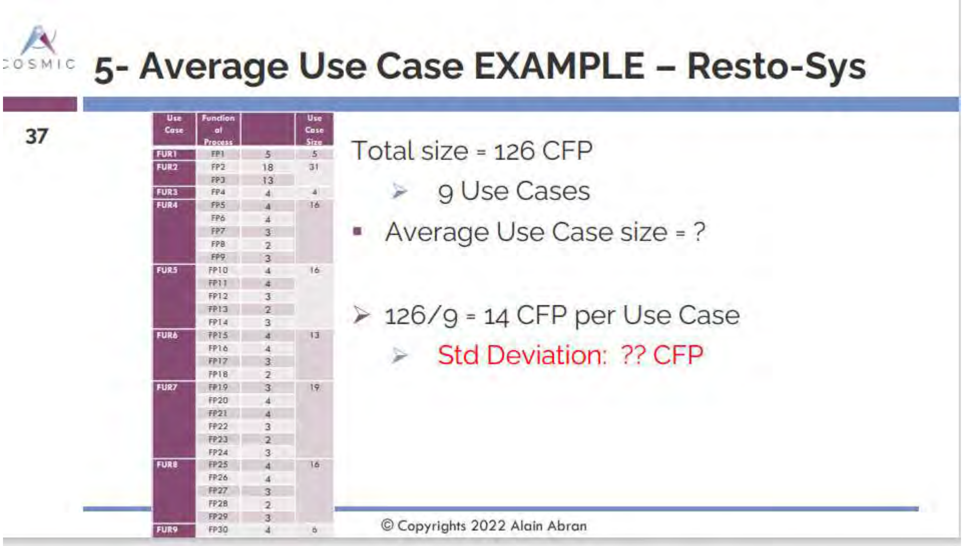
- 6 functional processes on average for a use case (Std Deviation: ?? FP)
- each functional process on average size = 8 CFP
  - Std Deviation: ?? CFP [range + or – 1 Std CFP]
- Hence the average size of a use case = ? = 8 x 6 = 48 CFP per use case.

For a new project with 12 use cases = ?

- software size would be 12 x 48 = 576 CFP.

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

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



COSMIC

5- Average Use Case EXAMPLE – Resto-Sys

37

Use Case	Function al Process		Use Case Size
FUR1	FP1	5	5
FUR2	FP2	18	31
	FP3	13	
FUR3	FP4	4	4
FUR4	FP5	4	16
	FP6	4	
	FP7	3	
	FP8	2	
	FP9	3	
FUR5	FP10	4	16
	FP11	4	
	FP12	3	
	FP13	2	
	FP14	3	
FUR6	FP15	4	13
	FP16	4	
	FP17	3	
	FP18	2	
FUR7	FP19	3	19
	FP20	4	
	FP21	4	
	FP22	3	
	FP23	2	
	FP24	3	
FUR8	FP25	4	16
	FP26	4	
	FP27	3	
	FP28	2	
	FP29	3	
FUR9	FP30	4	6

Total size = 126 CFP

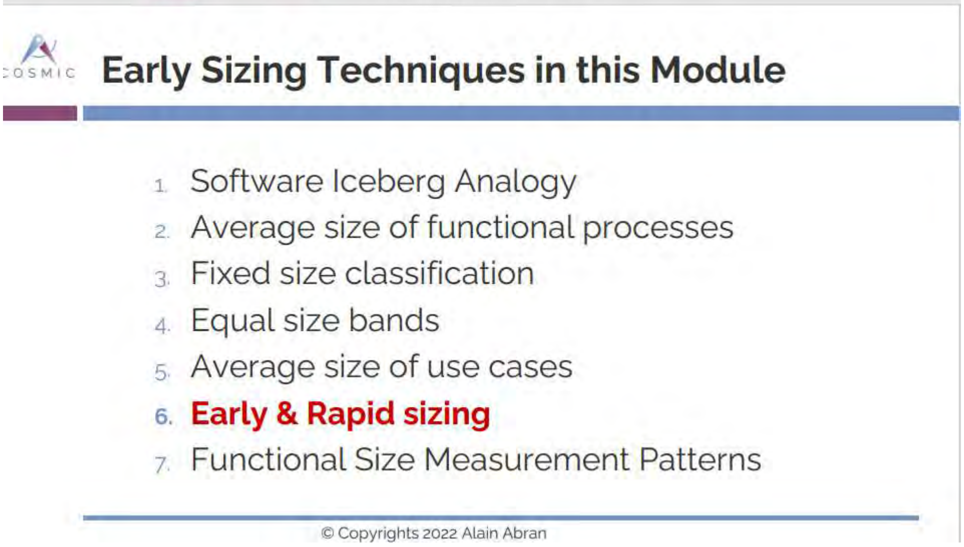
- ➢ 9 Use Cases

- ▪ Average Use Case size = ?

➢ 126/9 = 14 CFP per Use Case

- ➢ Std Deviation: ?? CFP

© Copyrights 2022 Alain Abran

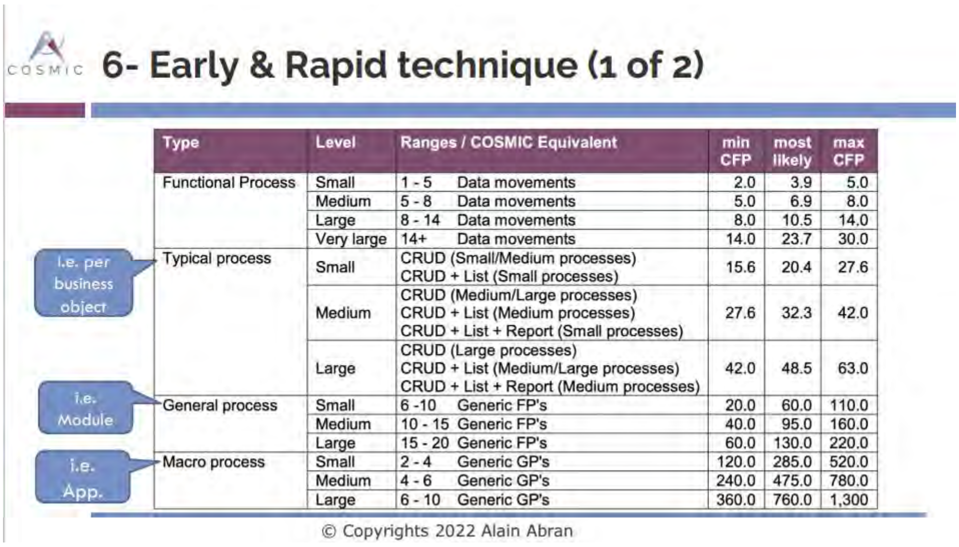


COSMIC

Early Sizing Techniques in this Module

1. Software Iceberg Analogy
2. Average size of functional processes
3. Fixed size classification
4. Equal size bands
5. Average size of use cases
6. **Early & Rapid sizing**
7. Functional Size Measurement Patterns

© Copyrights 2022 Alain Abran



COSMIC

6- Early & Rapid technique (1 of 2)

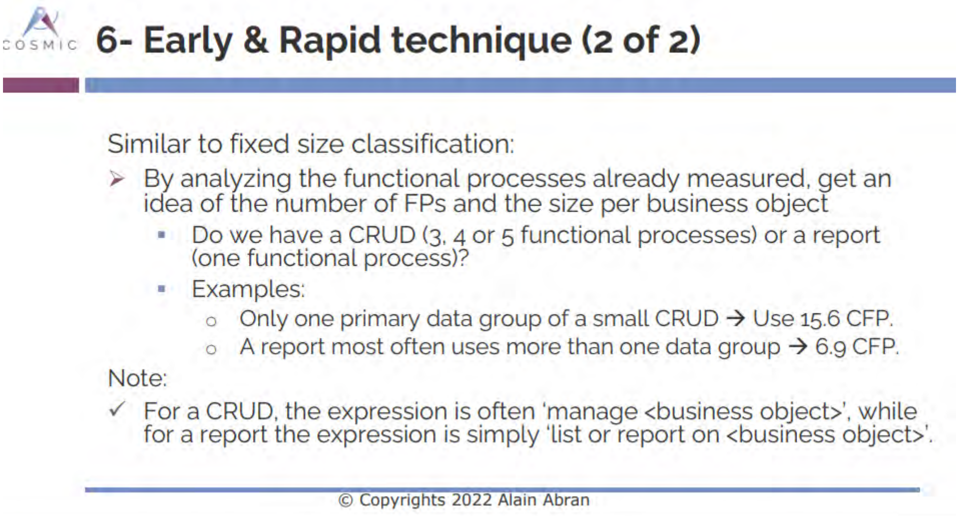
Type	Level	Ranges / COSMIC Equivalent	min CFP	most likely	max CFP
Functional Process	Small	1 - 5 Data movements	2.0	3.9	5.0
	Medium	5 - 8 Data movements	5.0	6.9	8.0
	Large	8 - 14 Data movements	8.0	10.5	14.0
	Very large	14+ Data movements	14.0	23.7	30.0
Typical process	Small	CRUD (Small/Medium processes) CRUD + List (Small processes)	15.6	20.4	27.6
	Medium	CRUD (Medium/Large processes) CRUD + List (Medium processes) CRUD + List + Report (Small processes)	27.6	32.3	42.0
	Large	CRUD (Large processes) CRUD + List (Medium/Large processes) CRUD + List + Report (Medium processes)	42.0	48.5	63.0
General process	Small	6 -10 Generic FP's	20.0	60.0	110.0
	Medium	10 - 15 Generic FP's	40.0	95.0	160.0
	Large	15 - 20 Generic FP's	60.0	130.0	220.0
Macro process	Small	2 - 4 Generic GP's	120.0	285.0	520.0
	Medium	4 - 6 Generic GP's	240.0	475.0	780.0
	Large	6 - 10 Generic GP's	360.0	760.0	1,300

i.e. per business object

i.e. Module

i.e. App.

© Copyrights 2022 Alain Abran



6- Early & Rapid technique (2 of 2)

Similar to fixed size classification:

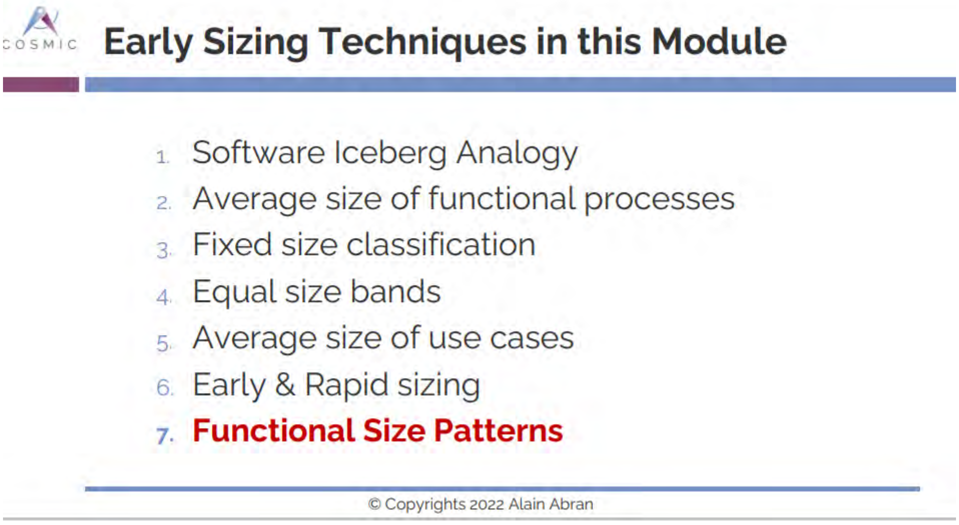
➢ By analyzing the functional processes already measured, get an idea of the number of FPs and the size per business object

- Do we have a CRUD (3, 4 or 5 functional processes) or a report (one functional process)?
- Examples:
  - Only one primary data group of a small CRUD → Use 15.6 CFP.
  - A report most often uses more than one data group → 6.9 CFP.

Note:

✓ For a CRUD, the expression is often 'manage <business object>', while for a report the expression is simply 'list or report on <business object>'.

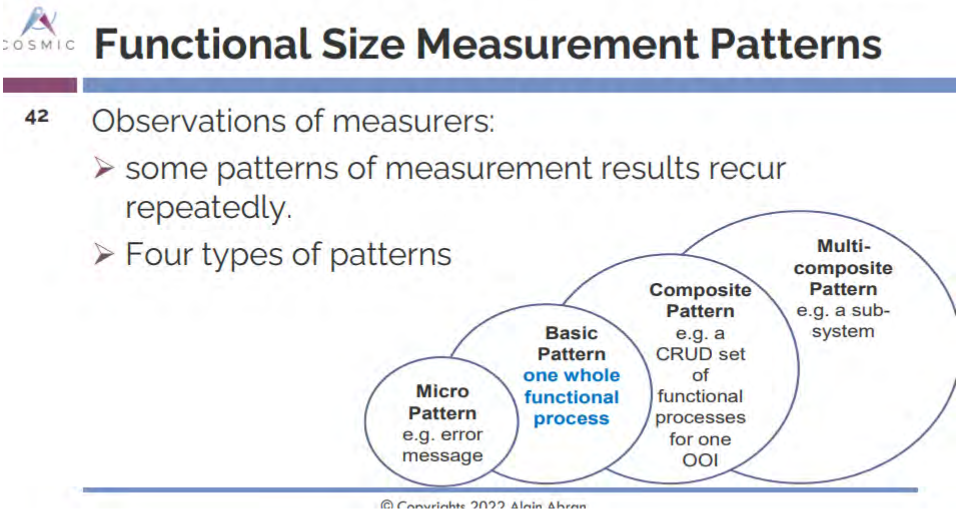
© Copyrights 2022 Alain Abran



Early Sizing Techniques in this Module

1. Software Iceberg Analogy
2. Average size of functional processes
3. Fixed size classification
4. Equal size bands
5. Average size of use cases
6. Early & Rapid sizing
7. **Functional Size Patterns**

© Copyrights 2022 Alain Abran



Functional Size Measurement Patterns

42

Observations of measurers:

➢ some patterns of measurement results recur repeatedly.

➢ Four types of patterns

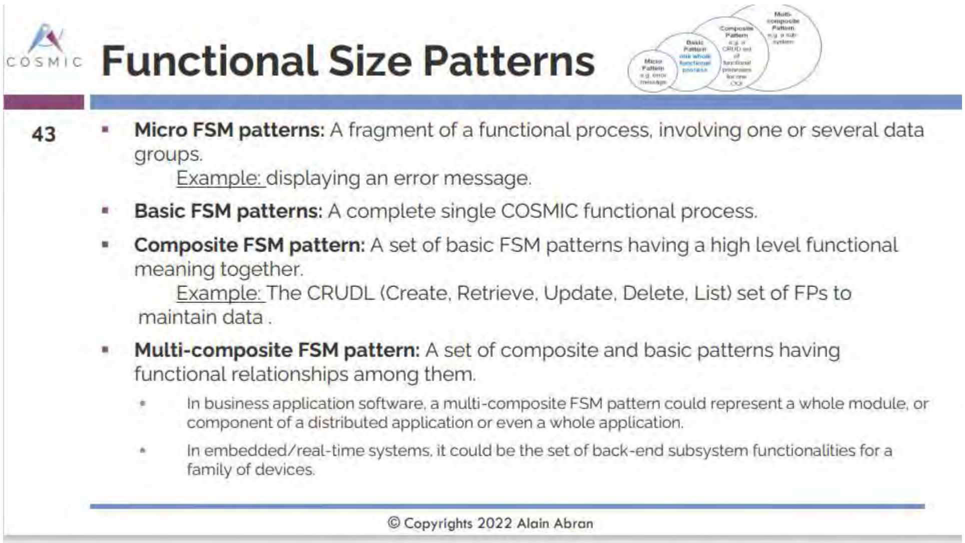
Micro Pattern e.g. error message

Basic Pattern one whole functional process

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

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

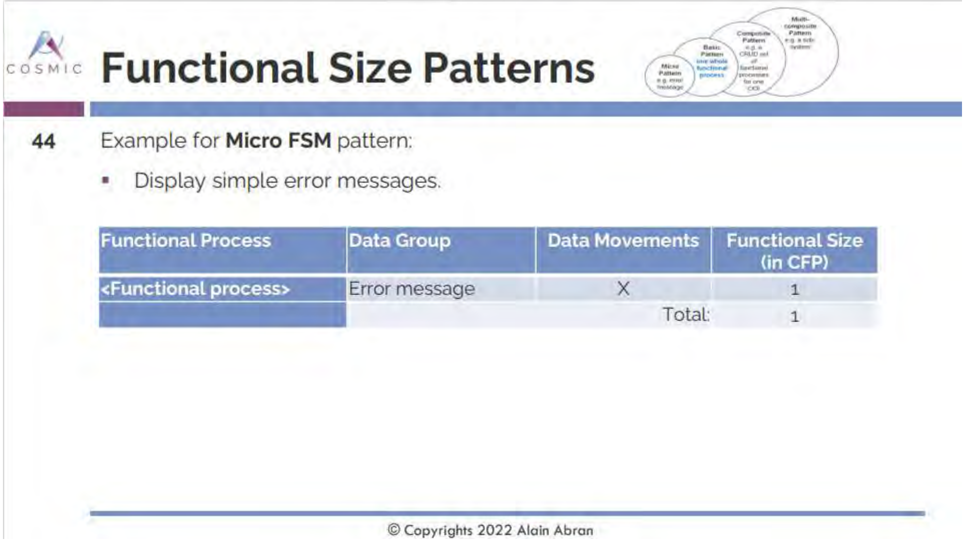
© Copyrights 2022 Alain Abran


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

Functional Size Patterns

43

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


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

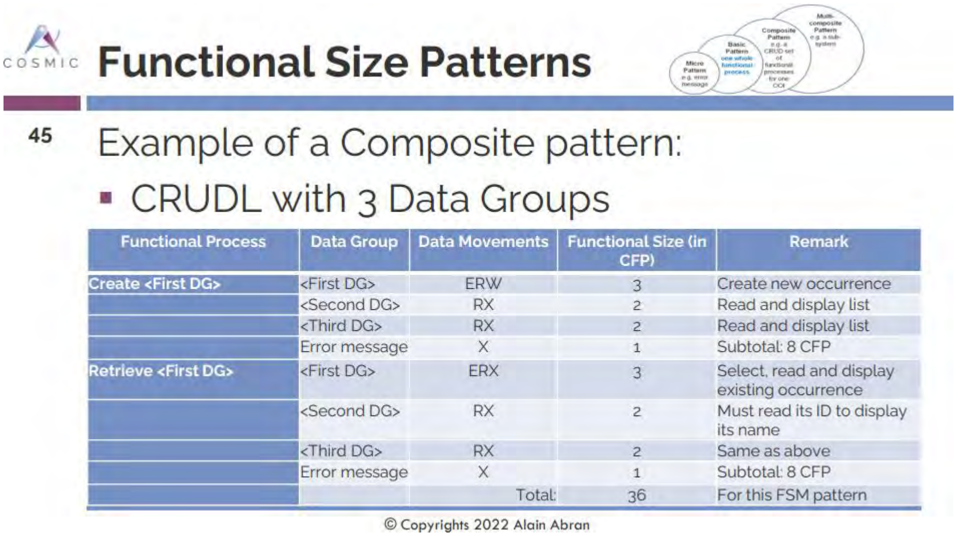
Functional Size Patterns

44

Example for **Micro FSM** pattern:

- Display simple error messages.

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


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

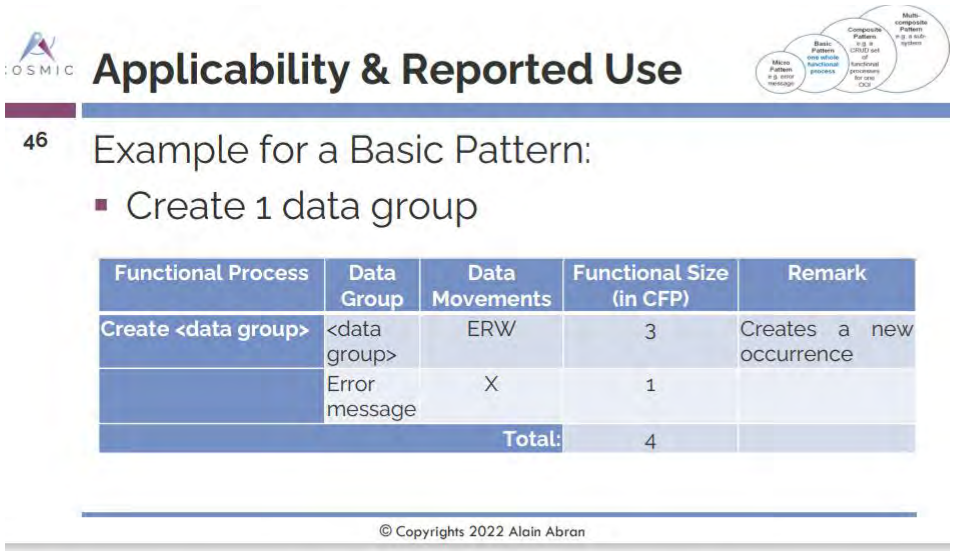
Functional Size Patterns

45

Example of a Composite pattern:

- CRUDL with 3 Data Groups

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


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

Applicability & Reported Use

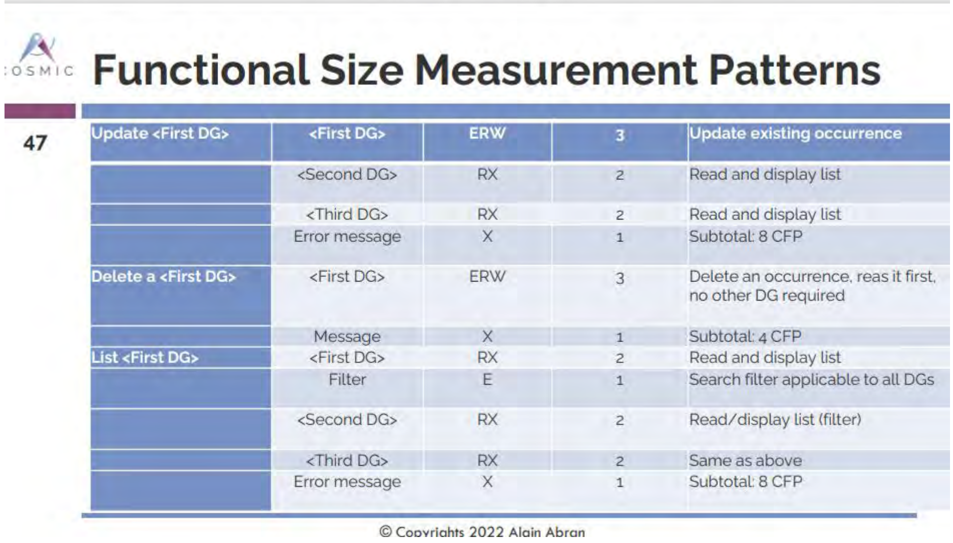
46

Example for a Basic Pattern:

- Create 1 data group

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

© Copyrights 2022 Alain Abran

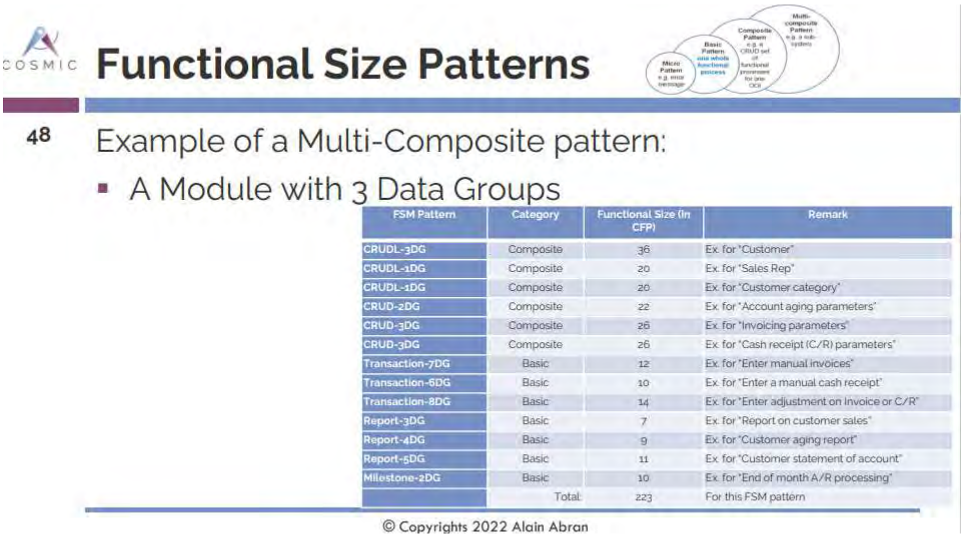


Functional Size Measurement Patterns

47

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

© Copyrights 2022 Alain Abran


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

Functional Size Patterns

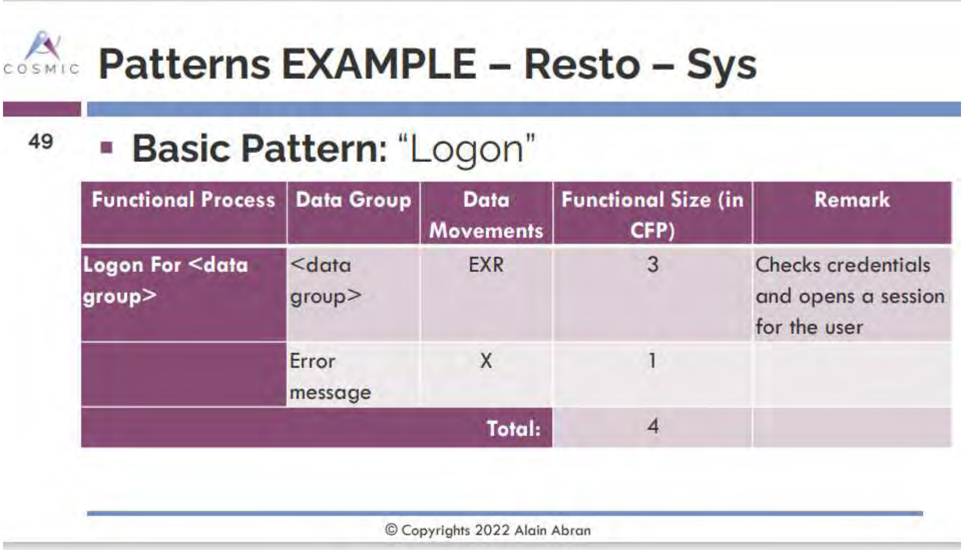
48

Example of a Multi-Composite pattern:

- A Module with 3 Data Groups

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

© Copyrights 2022 Alain Abran

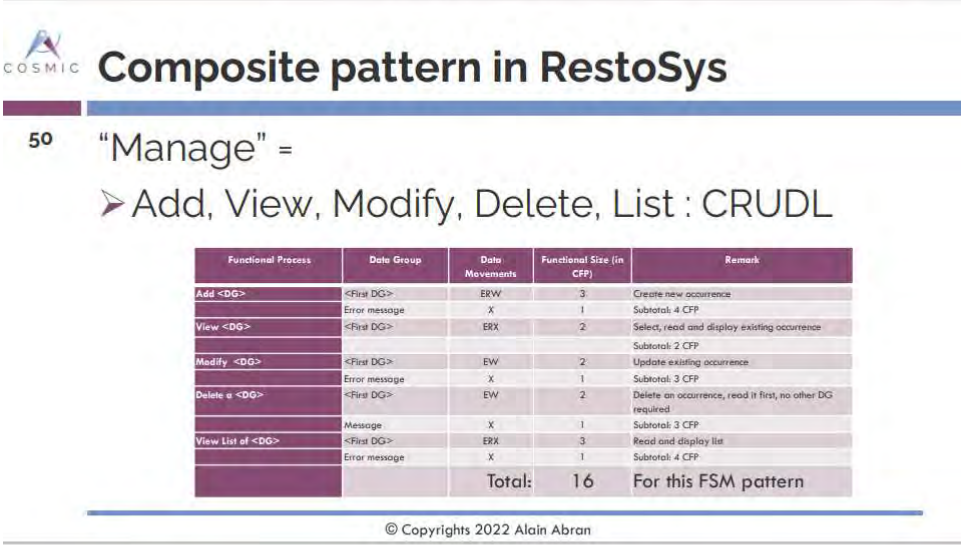


Patterns EXAMPLE – Resto – Sys

49

- **Basic Pattern:** "Logon"

Functional Process	Data Group	Data Movements	Functional Size (in CFP)	Remark
Logon For <data group>	<data group>	EXR	3	Checks credentials and opens a session for the user
	Error message	X	1	
		Total:	4	



Composite pattern in RestoSys

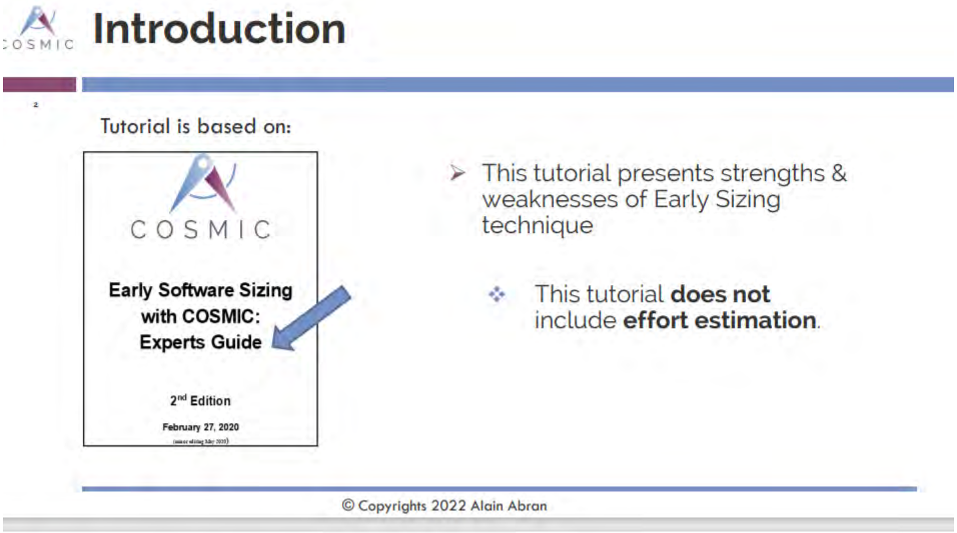
50

"Manage" =

➢Add, View, Modify, Delete, List : CRUDL

Functional Process	Data Group	Data Movements	Functional Size (in CFP)	Remark
Add <DG>	<First DG>	ERW	3	Create new occurrence
	Error message	X	1	Subtotal: 4 CFP
View <DG>	<First DG>	ERX	2	Select, read and display existing occurrence
				Subtotal: 2 CFP
Modify <DG>	<First DG>	EW	2	Update existing occurrence
	Error message	X	1	Subtotal: 3 CFP
Delete a <DG>	<First DG>	EW	2	Delete an occurrence, read it first, no other DG required
	Message	X	1	Subtotal: 3 CFP
View List of <DG>	<First DG>	ERX	3	Read and display list
	Error message	X	1	Subtotal: 4 CFP
		Total:	16	For this FSM pattern

MODULE 2: Selection of Techniques



Introduction

2

Tutorial is based on:

COSMIC

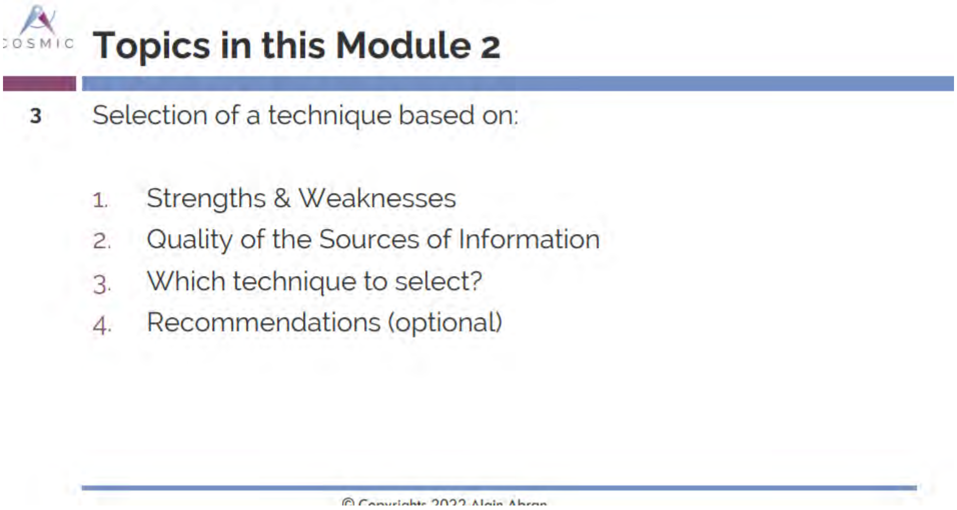
Early Software Sizing with COSMIC: Experts Guide

2nd Edition

February 27, 2020

- This tutorial presents strengths & weaknesses of Early Sizing technique
  - This tutorial **does not** include **effort estimation**.

© Copyrights 2022 Alain Abran

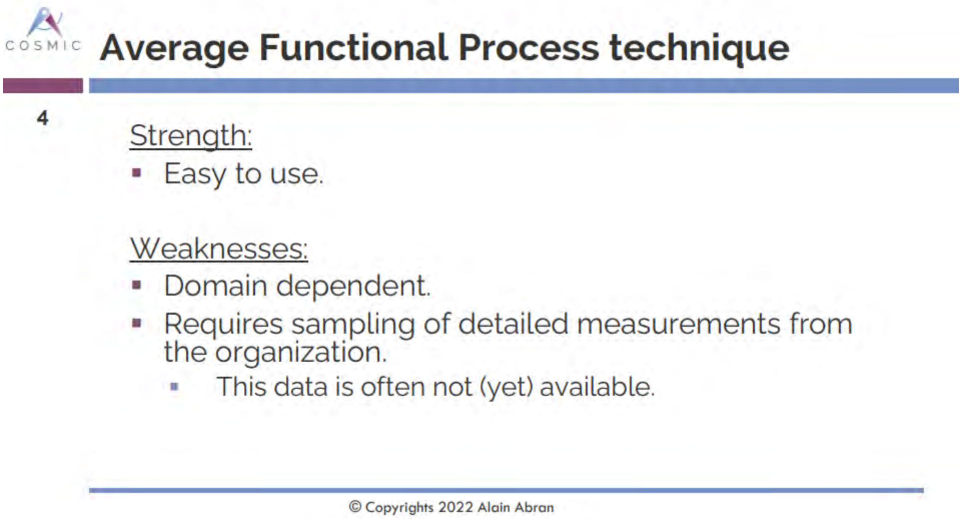


Topics in this Module 2

3 Selection of a technique based on:

1. Strengths & Weaknesses
2. Quality of the Sources of Information
3. Which technique to select?
4. Recommendations (optional)

© Copyrights 2022 Alain Abran



Average Functional Process technique

4

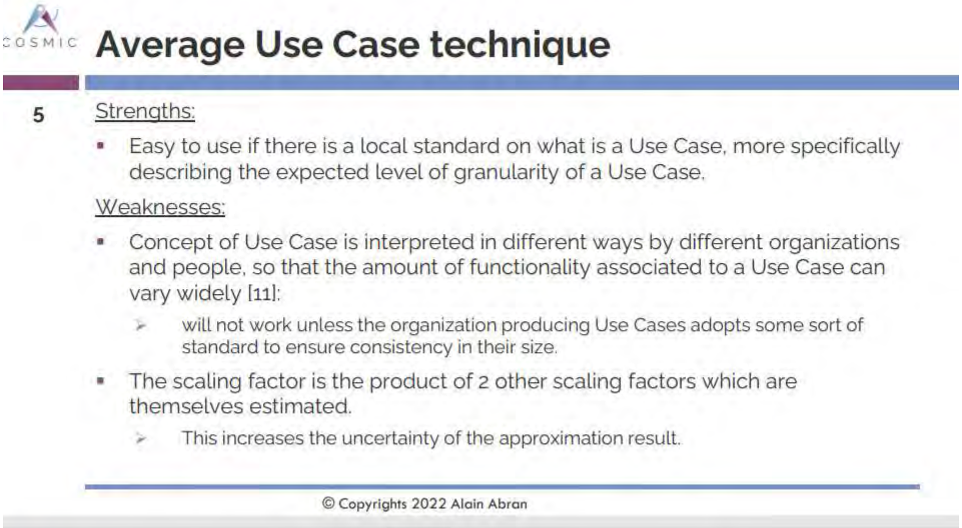
Strength:

- Easy to use.

Weaknesses:

- Domain dependent.
- Requires sampling of detailed measurements from the organization.
  - This data is often not (yet) available.

© Copyrights 2022 Alain Abran



Average Use Case technique

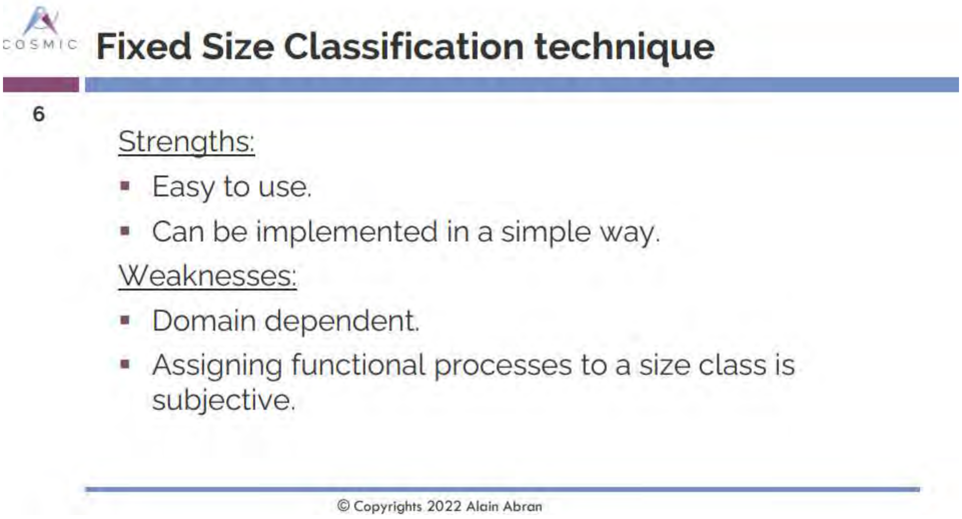
5

Strengths:

- Easy to use if there is a local standard on what is a Use Case, more specifically describing the expected level of granularity of a Use Case.

Weaknesses:

- Concept of Use Case is interpreted in different ways by different organizations and people, so that the amount of functionality associated to a Use Case can vary widely [11]:
  - will not work unless the organization producing Use Cases adopts some sort of standard to ensure consistency in their size.
- The scaling factor is the product of 2 other scaling factors which are themselves estimated.
  - This increases the uncertainty of the approximation result.



Fixed Size Classification technique

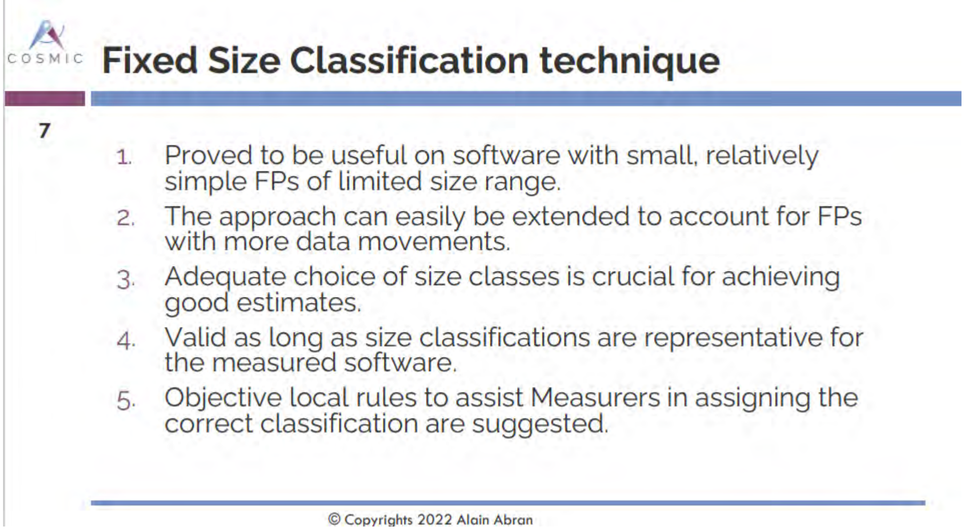
6

Strengths:

- Easy to use.
- Can be implemented in a simple way.

Weaknesses:

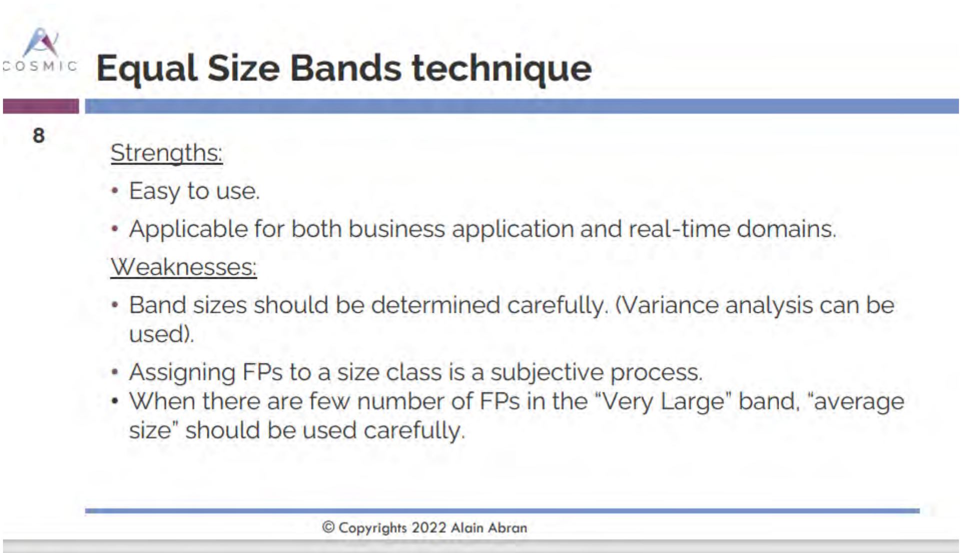
- Domain dependent.
- Assigning functional processes to a size class is subjective.



Fixed Size Classification technique

7

1. Proved to be useful on software with small, relatively simple FPs of limited size range.
2. The approach can easily be extended to account for FPs with more data movements.
3. Adequate choice of size classes is crucial for achieving good estimates.
4. Valid as long as size classifications are representative for the measured software.
5. Objective local rules to assist Measurers in assigning the correct classification are suggested.



Equal Size Bands technique

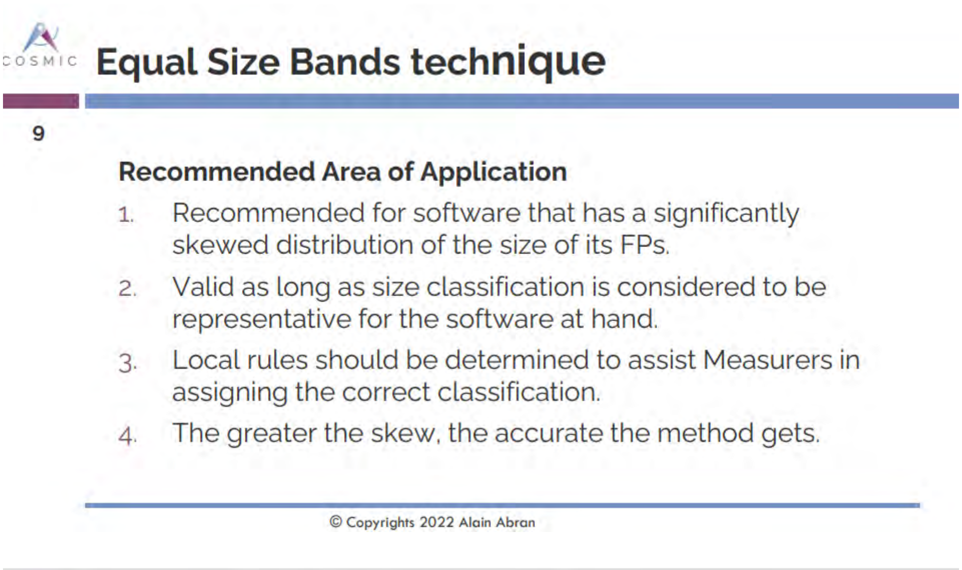
8

Strengths:

- Easy to use.
- Applicable for both business application and real-time domains.

Weaknesses:

- Band sizes should be determined carefully. (Variance analysis can be used).
- Assigning FPs to a size class is a subjective process.
- When there are few number of FPs in the "Very Large" band, "average size" should be used carefully.

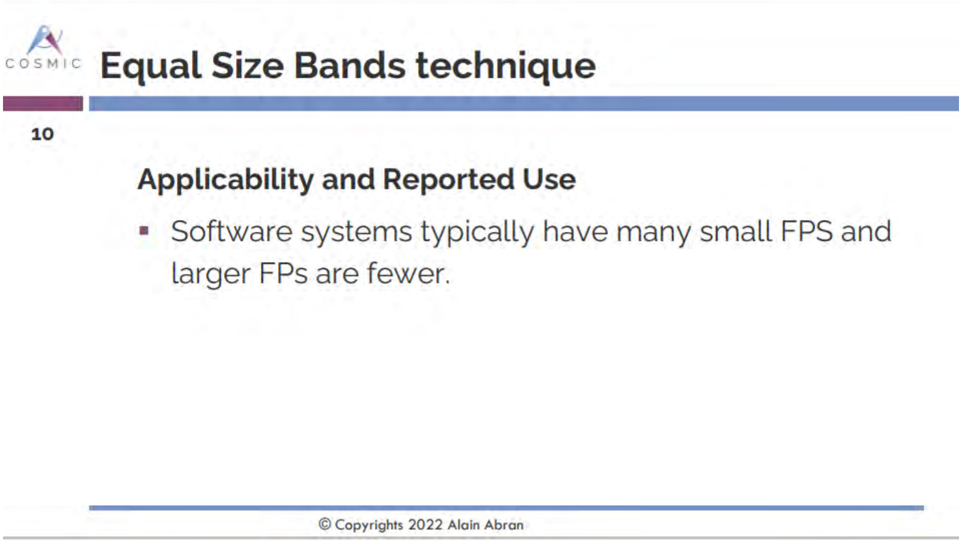


Equal Size Bands technique

9

Recommended Area of Application

1. Recommended for software that has a significantly skewed distribution of the size of its FPs.
2. Valid as long as size classification is considered to be representative for the software at hand.
3. Local rules should be determined to assist Measurers in assigning the correct classification.
4. The greater the skew, the accurate the method gets.

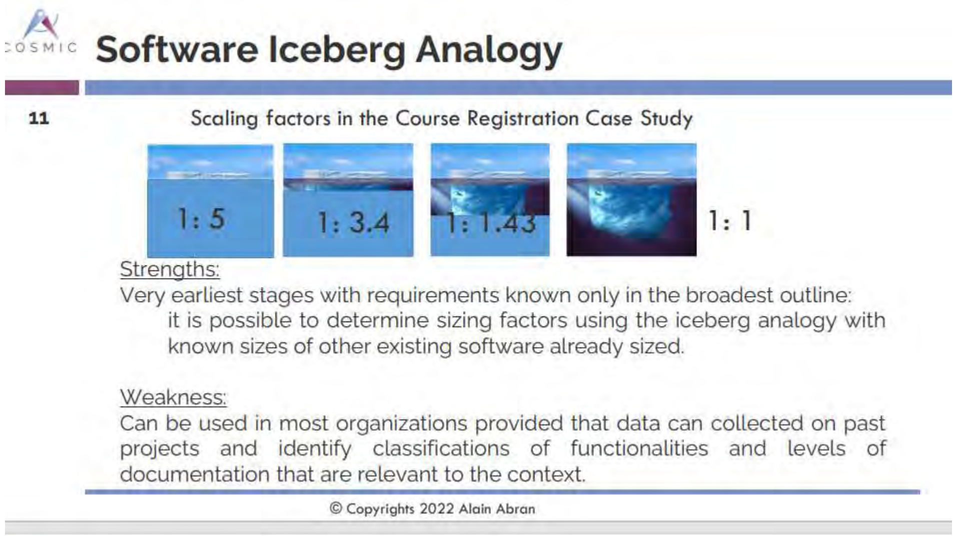


Equal Size Bands technique

10

Applicability and Reported Use

- Software systems typically have many small FPS and larger FPs are fewer.



COSMIC
1: 5
1: 3.4
1: 1.43
1: 1

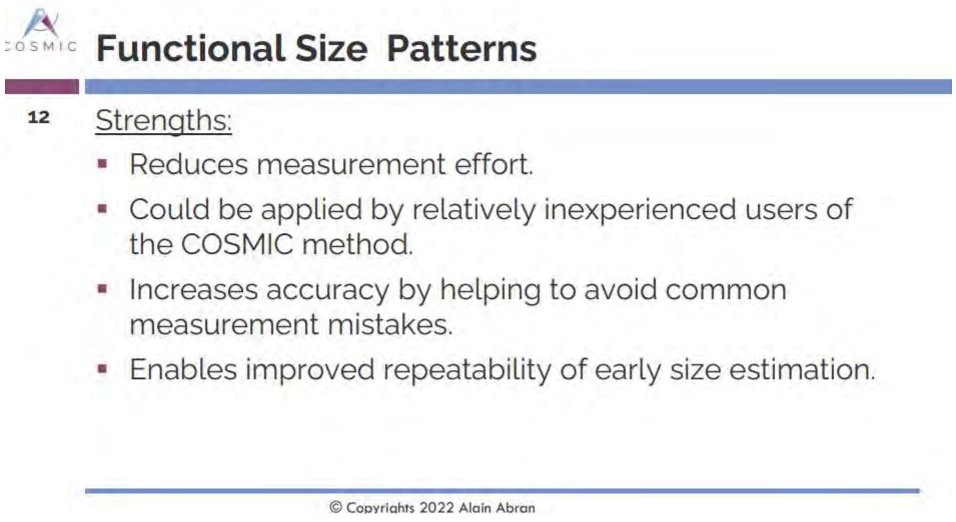
Software Iceberg Analogy

11

Scaling factors in the Course Registration Case Study

Strengths:
Very earliest stages with requirements known only in the broadest outline:
it is possible to determine sizing factors using the iceberg analogy with known sizes of other existing software already sized.

Weakness:
Can be used in most organizations provided that data can collected on past projects and identify classifications of functionalities and levels of documentation that are relevant to the context.

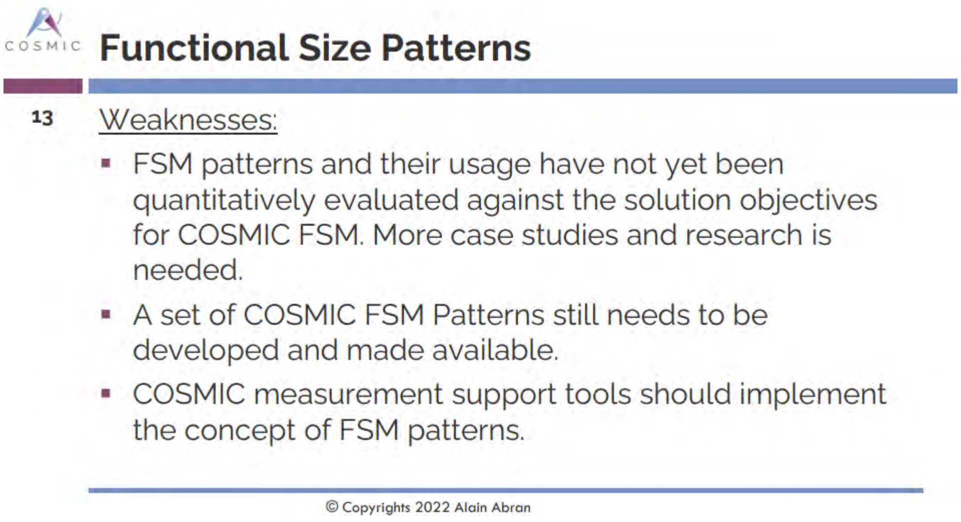


Functional Size Patterns

12

Strengths:

- Reduces measurement effort.
- Could be applied by relatively inexperienced users of the COSMIC method.
- Increases accuracy by helping to avoid common measurement mistakes.
- Enables improved repeatability of early size estimation.

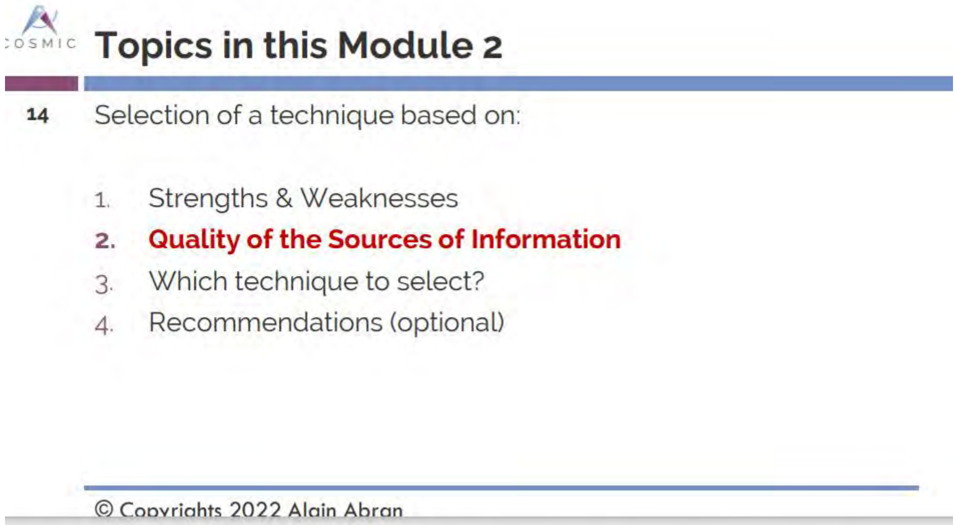


Functional Size Patterns

13

Weaknesses:

- FSM patterns and their usage have not yet been quantitatively evaluated against the solution objectives for COSMIC FSM. More case studies and research is needed.
- A set of COSMIC FSM Patterns still needs to be developed and made available.
- COSMIC measurement support tools should implement the concept of FSM patterns.



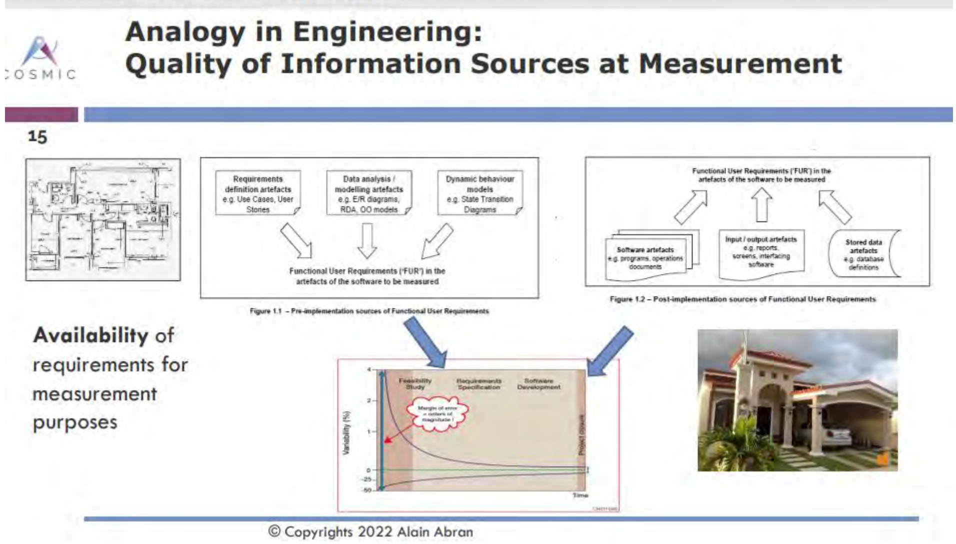
Topics in this Module 2

14

Selection of a technique based on:

1. Strengths & Weaknesses
2. **Quality of the Sources of Information**
3. Which technique to select?
4. Recommendations (optional)

© Copyrights 2022 Alain Abran



Analogy in Engineering: Quality of Information Sources at Measurement

15

Requirements definition artefacts e.g. Use Cases, User Stories
Data analysis / modelling artefacts e.g. E/R diagrams, RDA, OO models
Dynamic behaviour models e.g. State Transition Diagrams
Functional User Requirements ('FUR') in the artefacts of the software to be measured

Figure 1.1 – Pre-implementation sources of Functional User Requirements

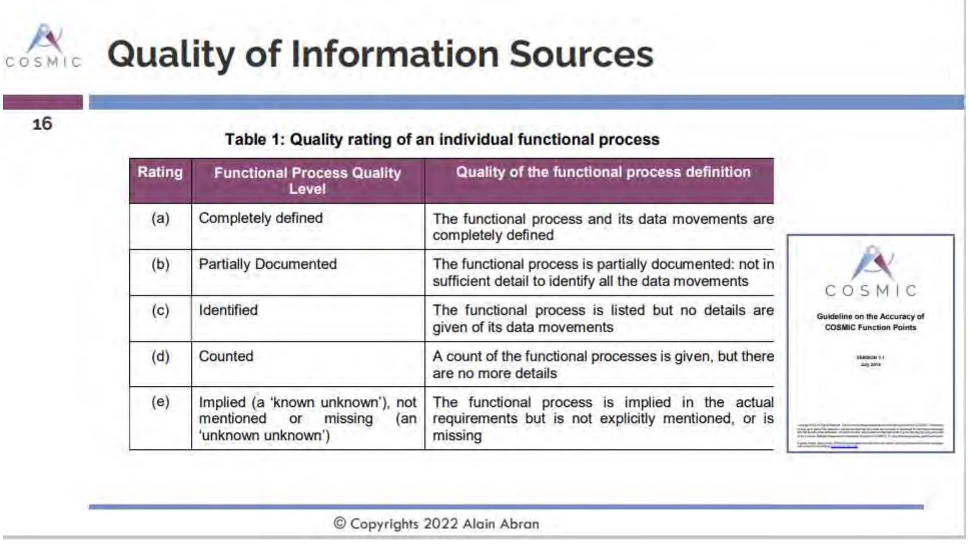
Functional User Requirements ('FUR') in the artefacts of the software to be measured
Software artefacts e.g. programs, operations documents
Input / output artefacts e.g. reports, screens, interfacing software
Stored data artefacts e.g. database definitions

Figure 1.2 – Post-implementation sources of Functional User Requirements

Availability of requirements for measurement purposes

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

© Copyrights 2022 Alain Abran



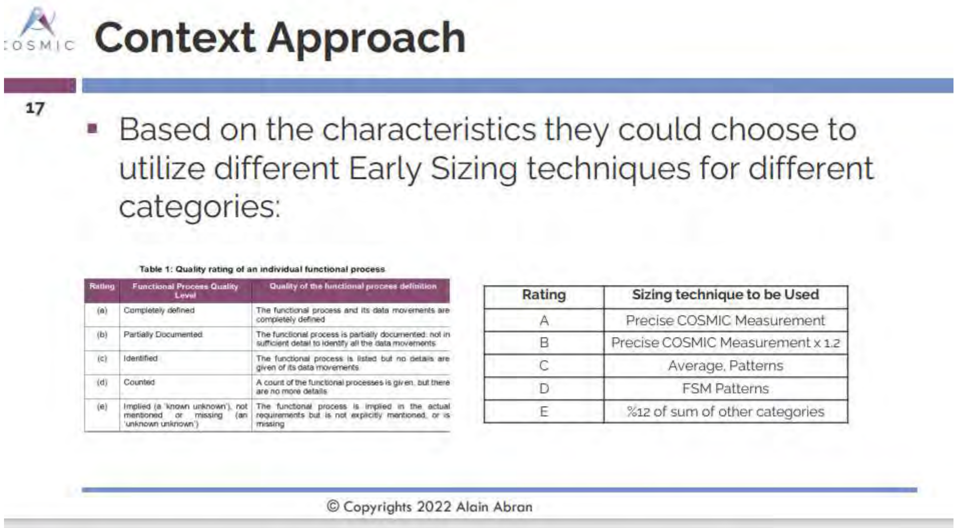
Quality of Information Sources

16

Table 1: Quality rating of an individual functional process

Rating	Functional Process Quality Level	Quality of the functional process definition
(a)	Completely defined	The functional process and its data movements are completely defined
(b)	Partially Documented	The functional process is partially documented: not in sufficient detail to identify all the data movements
(c)	Identified	The functional process is listed but no details are given of its data movements
(d)	Counted	A count of the functional processes is given, but there are no more details
(e)	Implied (a 'known unknown'), not mentioned or missing (an 'unknown unknown')	The functional process is implied in the actual requirements but is not explicitly mentioned, or is missing

© Copyrights 2022 Alain Abran



Context Approach

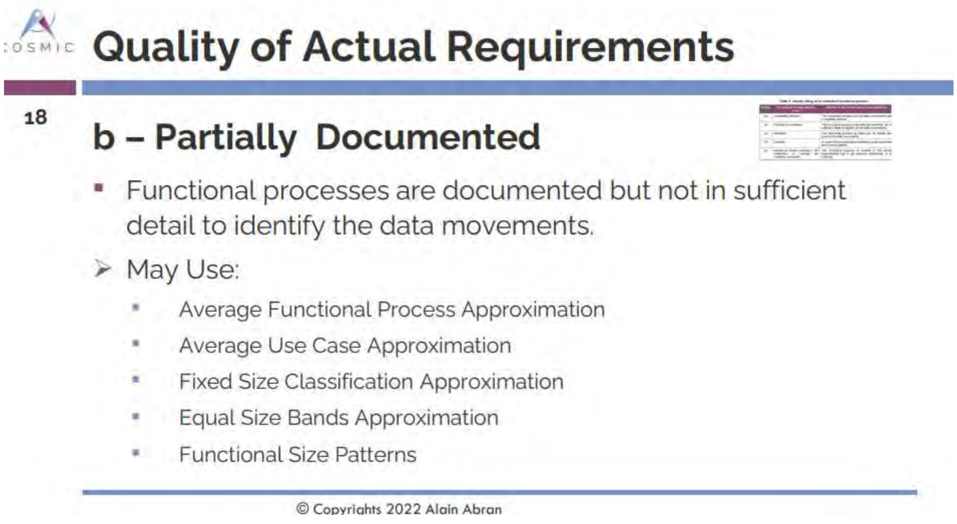
17

- Based on the characteristics they could choose to utilize different Early Sizing techniques for different categories:

Table 1: Quality rating of an individual functional process

Rating	Functional Process Quality Level	Quality of the functional process definition
(a)	Completely defined	The functional process and its data movements are completely defined
(b)	Partially Documented	The functional process is partially documented, not in sufficient detail to identify all the data movements
(c)	Identified	The functional process is listed but no details are given of its data movements
(d)	Counted	A count of the functional processes is given, but there are no more details
(e)	Implied (a 'known unknown'), not mentioned or missing (an 'unknown unknown')	The functional process is implied in the actual requirements but is not explicitly mentioned, or is missing

Rating	Sizing technique to be Used
A	Precise COSMIC Measurement
B	Precise COSMIC Measurement x 1.2
C	Average, Patterns
D	FSM Patterns
E	%12 of sum of other categories



Quality of Actual Requirements

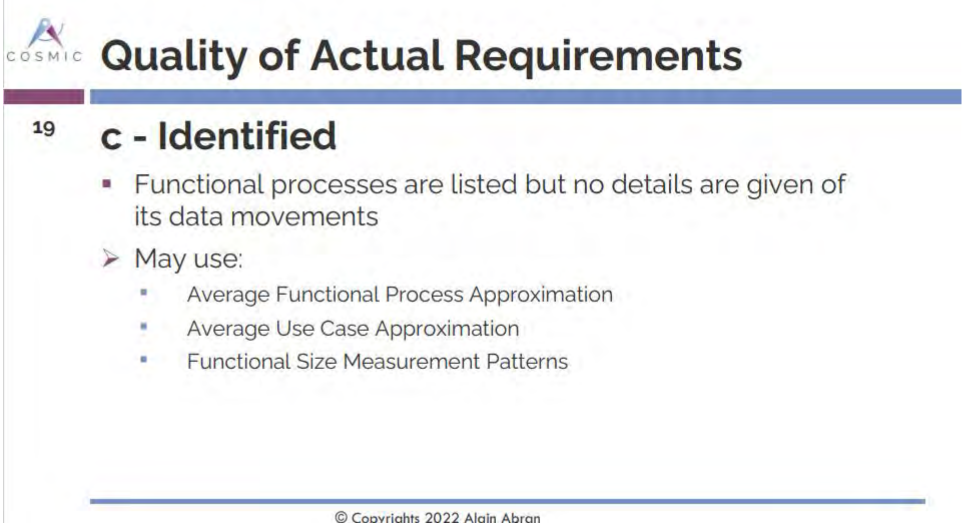
18

b – Partially Documented

- Functional processes are documented but not in sufficient detail to identify the data movements.

➢ May Use:

- Average Functional Process Approximation
- Average Use Case Approximation
- Fixed Size Classification Approximation
- Equal Size Bands Approximation
- Functional Size Patterns



Quality of Actual Requirements

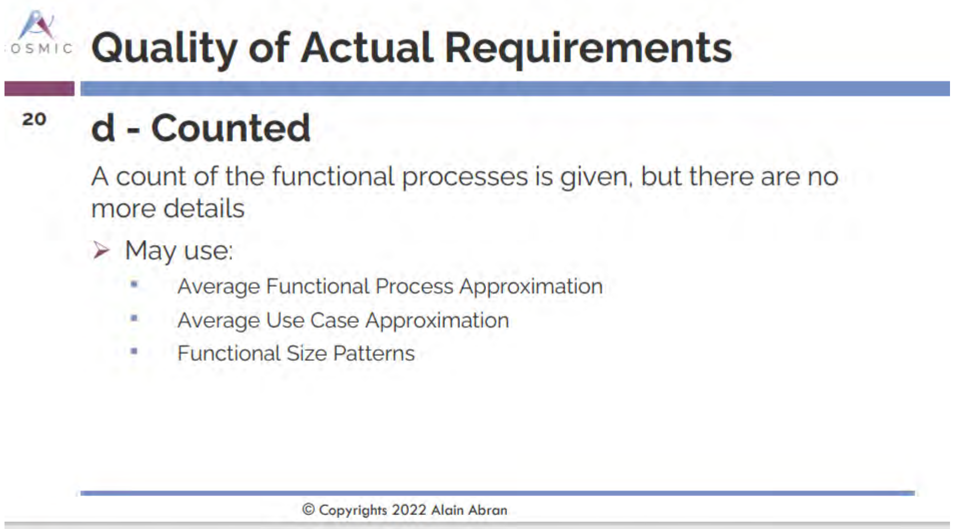
19

c - Identified

- Functional processes are listed but no details are given of its data movements

➢ May use:

- Average Functional Process Approximation
- Average Use Case Approximation
- Functional Size Measurement Patterns



Quality of Actual Requirements

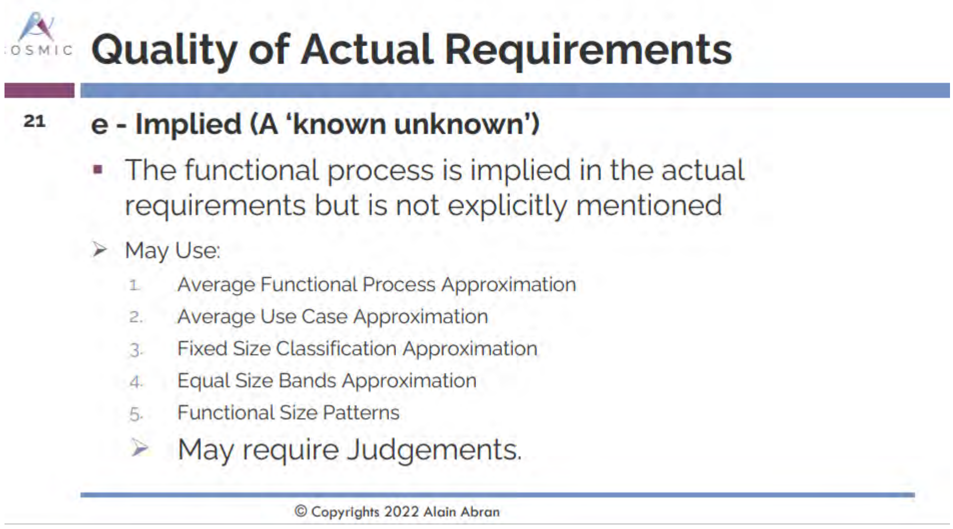
20

d - Counted

A count of the functional processes is given, but there are no more details

- May use:
  - Average Functional Process Approximation
  - Average Use Case Approximation
  - Functional Size Patterns

© Copyrights 2022 Alain Abran



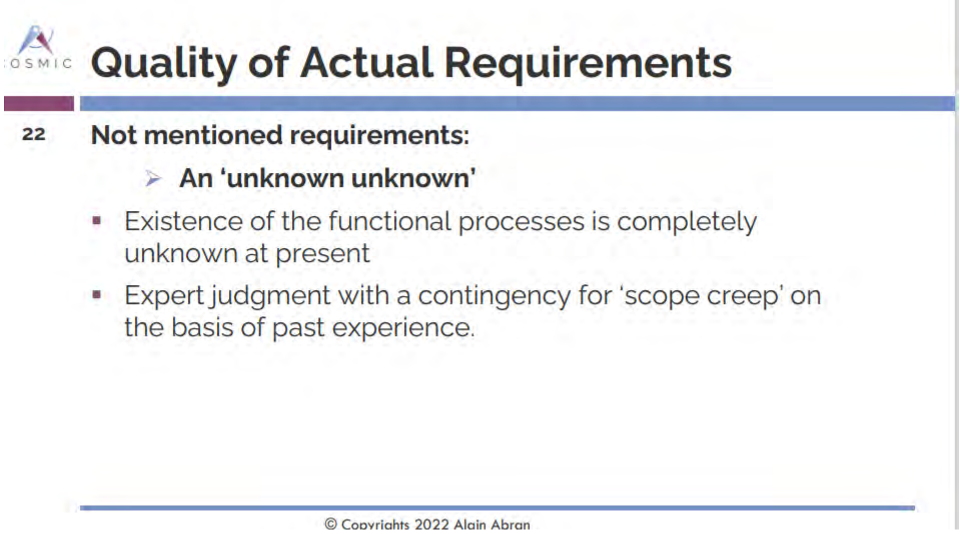
Quality of Actual Requirements

21

e - Implied (A 'known unknown')

- The functional process is implied in the actual requirements but is not explicitly mentioned
- May Use:
  1. Average Functional Process Approximation
  2. Average Use Case Approximation
  3. Fixed Size Classification Approximation
  4. Equal Size Bands Approximation
  5. Functional Size Patterns
  - May require Judgements.

© Copyrights 2022 Alain Abran



Quality of Actual Requirements

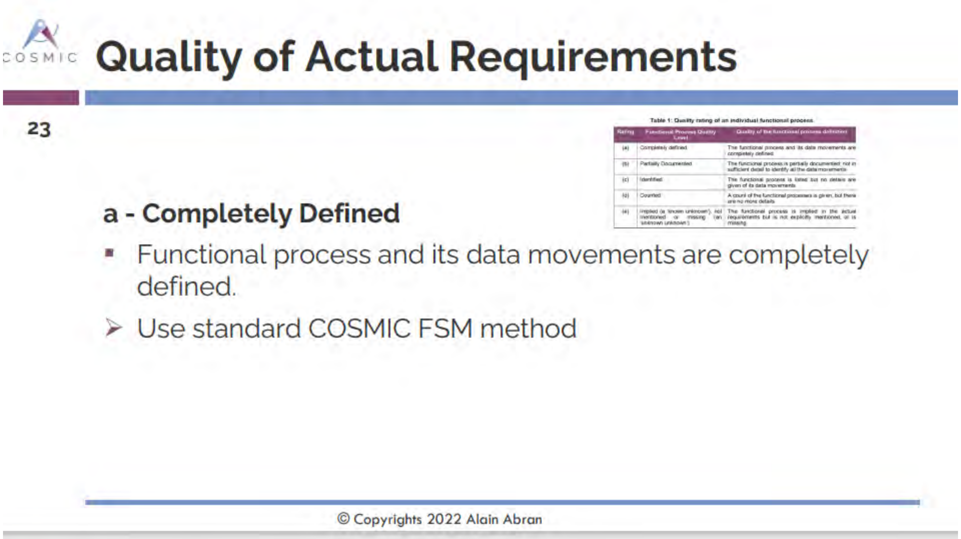
22

Not mentioned requirements:

- **An 'unknown unknown'**

- Existence of the functional processes is completely unknown at present
- Expert judgment with a contingency for 'scope creep' on the basis of past experience.

© Copyrights 2022 Alain Abran



Quality of Actual Requirements

23

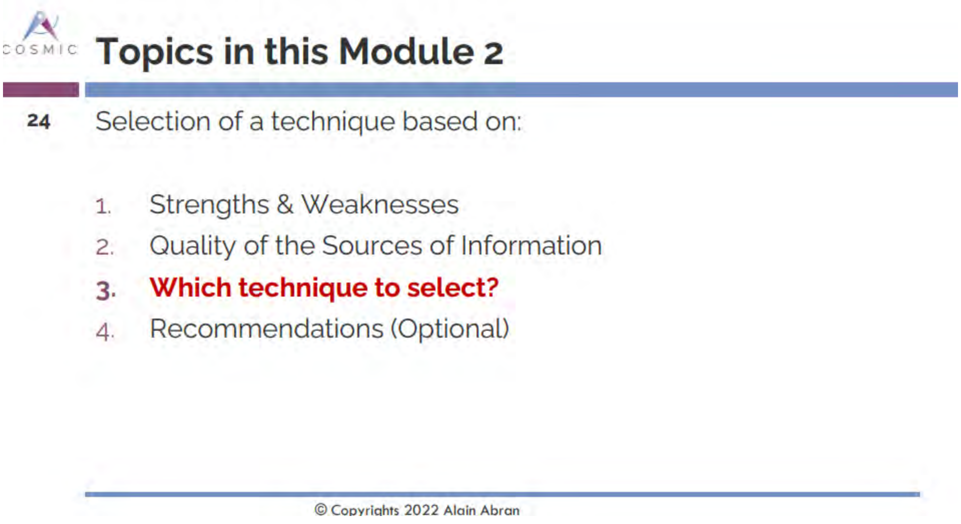
Table 1: Quality rating of an individual functional process

Rating	Functional Process Quality Level	Quality of the functional process definition
(a)	Completely defined	The functional process and its data movements are completely defined
(b)	Partially Documented	The functional process is partially documented: not in sufficient detail to identify all the data movements
(c)	Identified	The functional process is listed but no details are given of its data movements
(d)	Counted	A count of the functional processes is given, but there are no more details
(e)	Implied (a 'known unknown'), not mentioned or missing (an 'unknown unknown')	The functional process is implied in the actual requirements but is not explicitly mentioned, or is missing

a - Completely Defined

- Functional process and its data movements are completely defined.
- ➢ Use standard COSMIC FSM method

© Copyrights 2022 Alain Abran



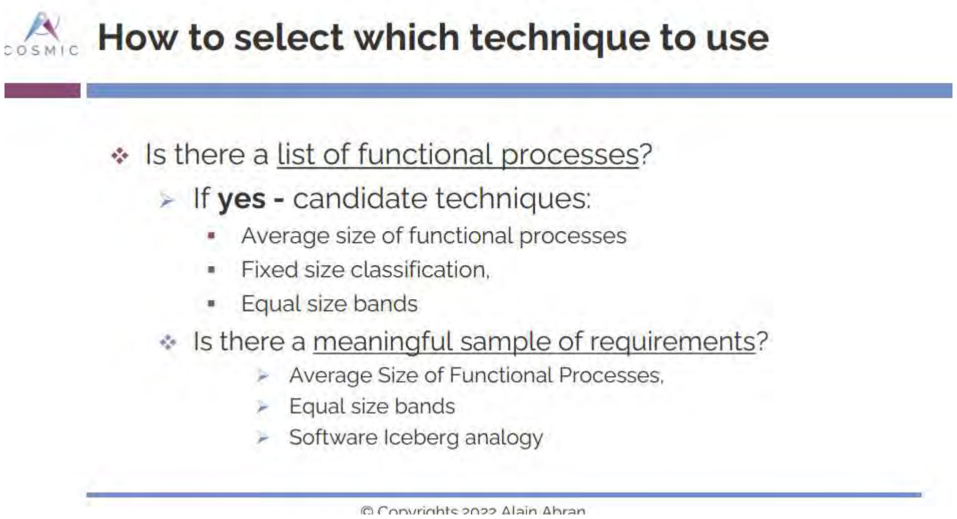
Topics in this Module 2

24

Selection of a technique based on:

1. Strengths & Weaknesses
2. Quality of the Sources of Information
3. **Which technique to select?**
4. Recommendations (Optional)

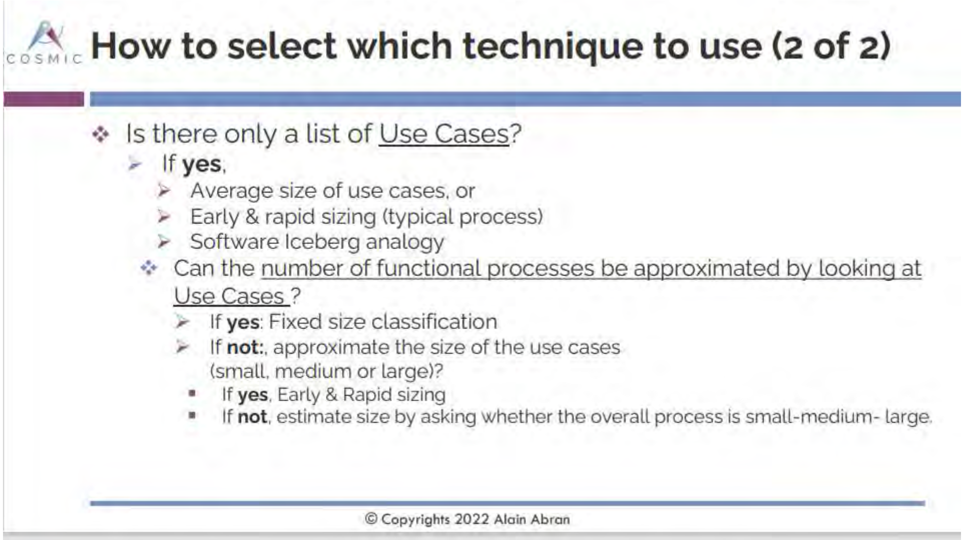
© Copyrights 2022 Alain Abran



How to select which technique to use

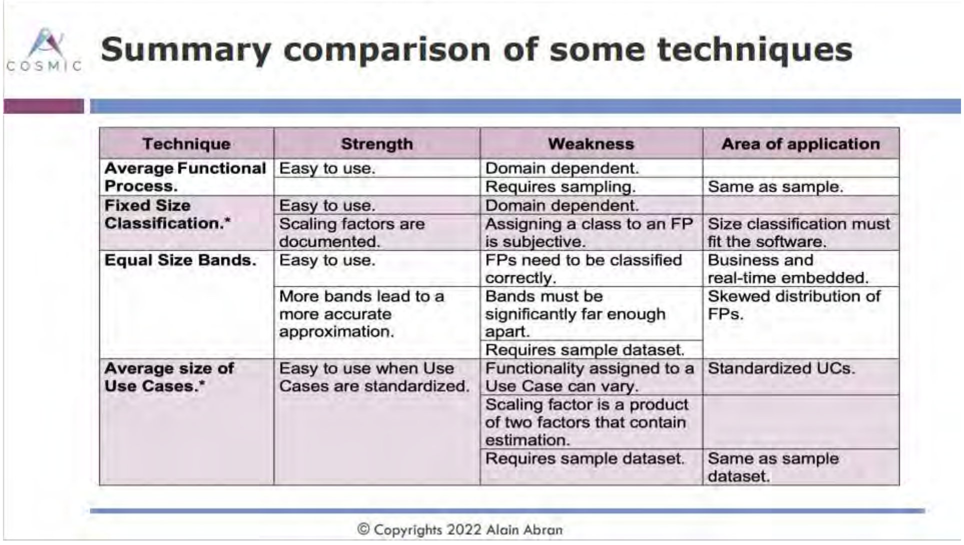
- ❖ Is there a list of functional processes?
  - ➢ If **yes -** candidate techniques:
    - Average size of functional processes
    - Fixed size classification,
    - Equal size bands
  - ❖ Is there a meaningful sample of requirements?
    - ➢ Average Size of Functional Processes,
    - ➢ Equal size bands
    - ➢ Software Iceberg analogy

© Copyrights 2022 Alain Abran



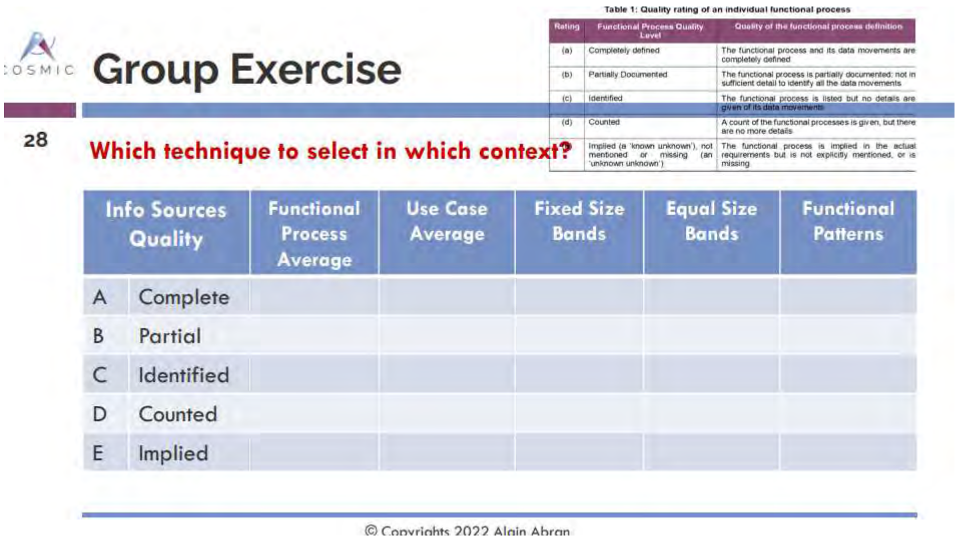
How to select which technique to use (2 of 2)

- Is there only a list of Use Cases?
  - If **yes**,
    - Average size of use cases, or
    - Early & rapid sizing (typical process)
    - Software Iceberg analogy
  - Can the number of functional processes be approximated by looking at Use Cases ?
    - If **yes**: Fixed size classification
    - If **not**:, approximate the size of the use cases (small, medium or large)?
      - If **yes**, Early & Rapid sizing
      - If **not**, estimate size by asking whether the overall process is small-medium- large.



Summary comparison of some techniques

Technique	Strength	Weakness	Area of application
Average Functional Process.	Easy to use.	Domain dependent.	
		Requires sampling.	Same as sample.
Fixed Size Classification.*	Easy to use.	Domain dependent.	
	Scaling factors are documented.	Assigning a class to an FP is subjective.	Size classification must fit the software.
Equal Size Bands.	Easy to use.	FPs need to be classified correctly.	Business and real-time embedded.
	More bands lead to a more accurate approximation.	Bands must be significantly far enough apart.	Skewed distribution of FPs.
		Requires sample dataset.	
Average size of Use Cases.*	Easy to use when Use Cases are standardized.	Functionality assigned to a Use Case can vary.	Standardized UCs.
		Scaling factor is a product of two factors that contain estimation.	
		Requires sample dataset.	Same as sample dataset.



Group Exercise

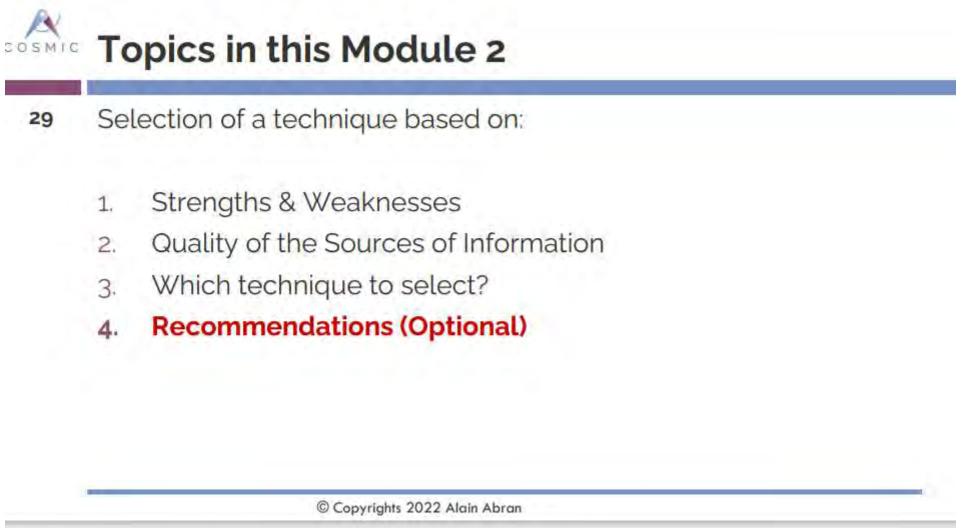
28

Which technique to select in which context?

Table 1: Quality rating of an individual functional process

Rating	Functional Process Quality Level	Quality of the functional process definition
(a)	Completely defined	The functional process and its data movements are completely defined
(b)	Partially Documented	The functional process is partially documented: not in sufficient detail to identify all the data movements
(c)	Identified	The functional process is listed but no details are given of its data movements
(d)	Counted	A count of the functional processes is given, but there are no more details
	Implied (a 'known unknown'), not mentioned or missing (an 'unknown unknown')	The functional process is implied in the actual requirements but is not explicitly mentioned, or is missing

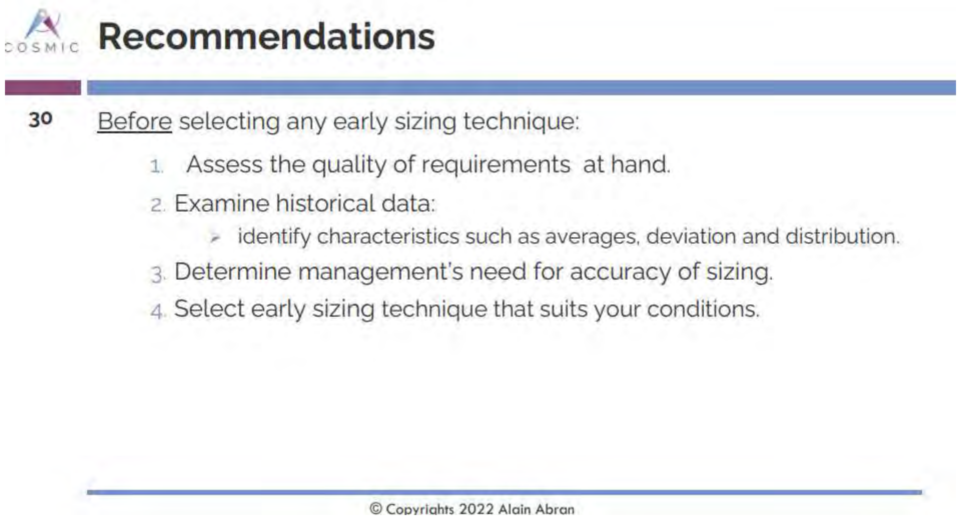
Info Sources Quality		Functional Process Average	Use Case Average	Fixed Size Bands	Equal Size Bands	Functional Patterns
A	Complete					
B	Partial					
C	Identified					
D	Counted					
E	Implied					



Topics in this Module 2

29 Selection of a technique based on:

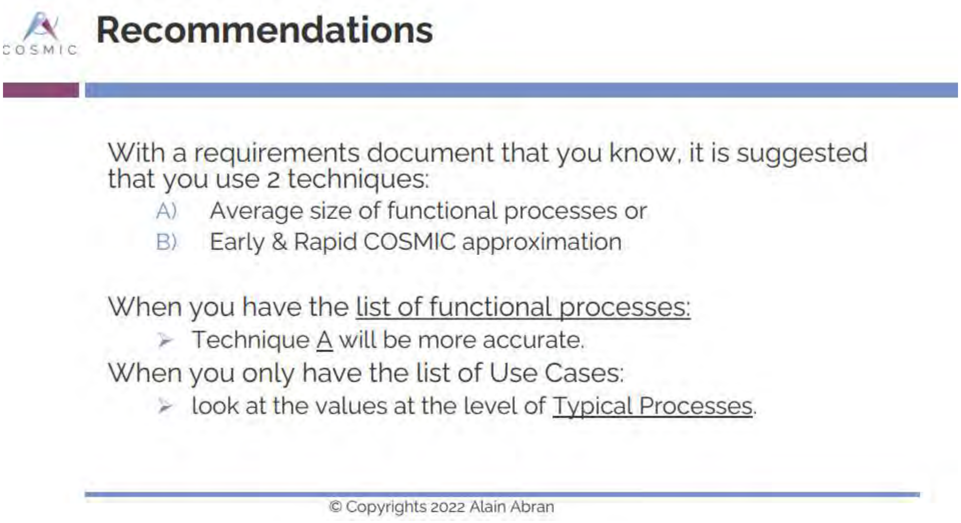
1. Strengths & Weaknesses
2. Quality of the Sources of Information
3. Which technique to select?
4. **Recommendations (Optional)**



Recommendations

30 Before selecting any early sizing technique:

1. Assess the quality of requirements at hand.
2. Examine historical data:
   - identify characteristics such as averages, deviation and distribution.
3. Determine management's need for accuracy of sizing.
4. Select early sizing technique that suits your conditions.



Recommendations

With a requirements document that you know, it is suggested that you use 2 techniques:

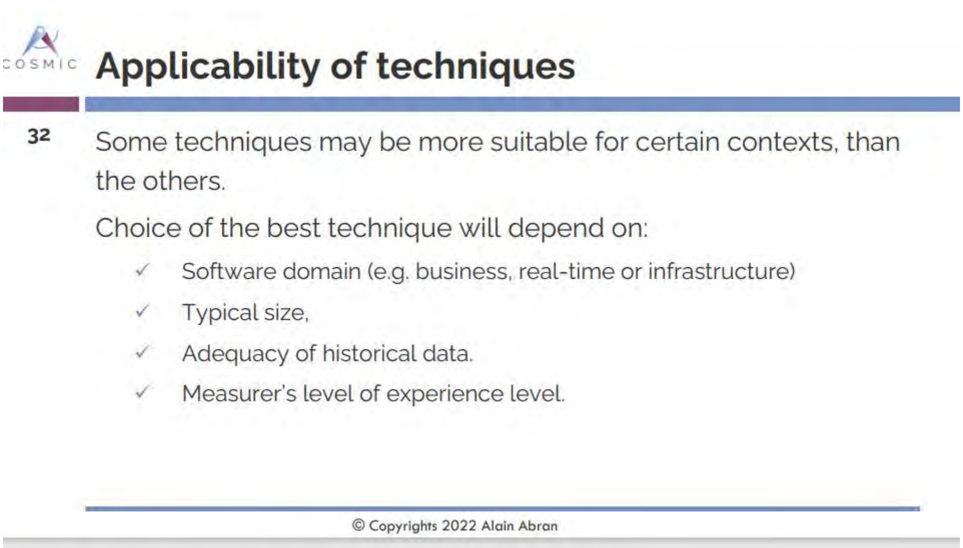
A) Average size of functional processes or
B) Early & Rapid COSMIC approximation

When you have the list of functional processes:

- Technique A will be more accurate.

When you only have the list of Use Cases:

- look at the values at the level of Typical Processes.



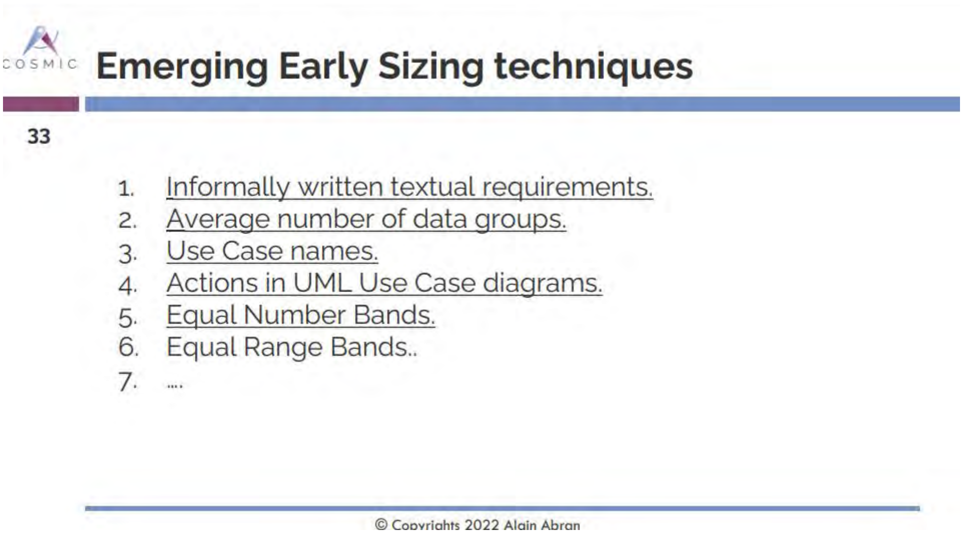
Applicability of techniques

32

Some techniques may be more suitable for certain contexts, than the others.

Choice of the best technique will depend on:

- ✓ Software domain (e.g. business, real-time or infrastructure)
- ✓ Typical size,
- ✓ Adequacy of historical data.
- ✓ Measurer's level of experience level.

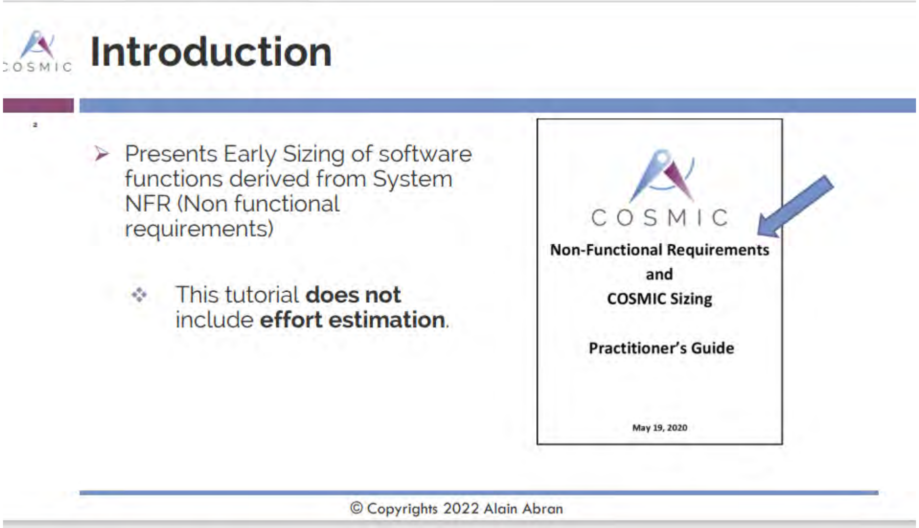


Emerging Early Sizing techniques

33

1. Informally written textual requirements.
2. Average number of data groups.
3. Use Case names.
4. Actions in UML Use Case diagrams.
5. Equal Number Bands.
6. Equal Range Bands..
7. ....

MODULE 3: Early Sizing & Non-Functional Requirements (NFR)



Introduction

2

- Presents Early Sizing of software functions derived from System NFR (Non functional requirements)
  - This tutorial **does not** include **effort estimation**.

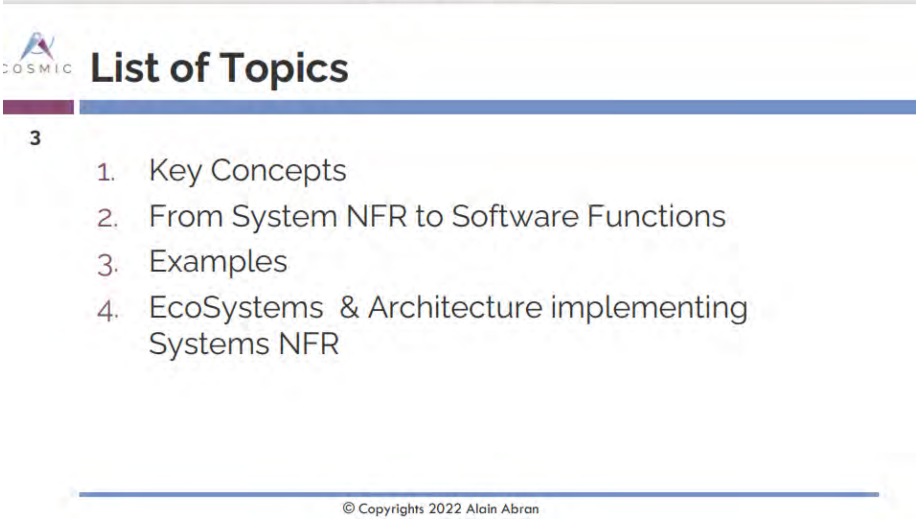
COSMIC

Non-Functional Requirements
and
COSMIC Sizing

Practitioner's Guide

May 19, 2020

© Copyrights 2022 Alain Abran

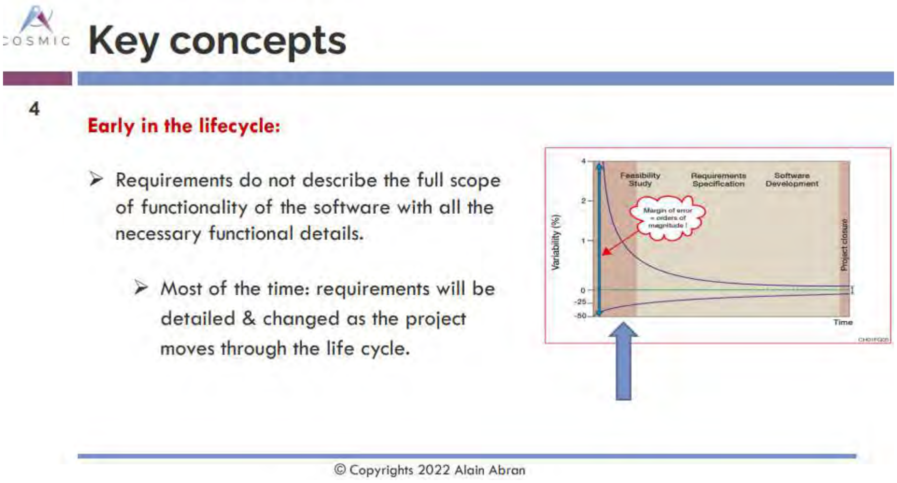


List of Topics

3

1. Key Concepts
2. From System NFR to Software Functions
3. Examples
4. EcoSystems & Architecture implementing Systems NFR

© Copyrights 2022 Alain Abran



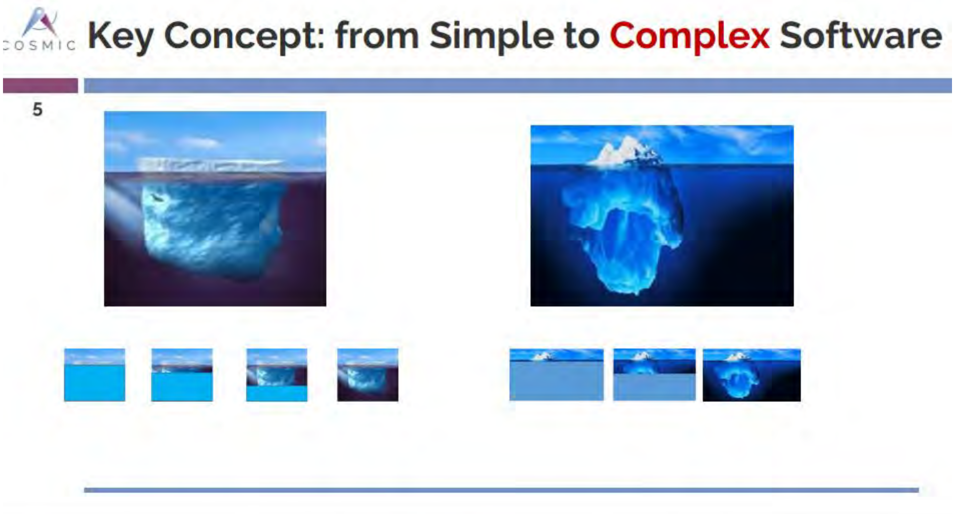
Key concepts

4

Early in the lifecycle:

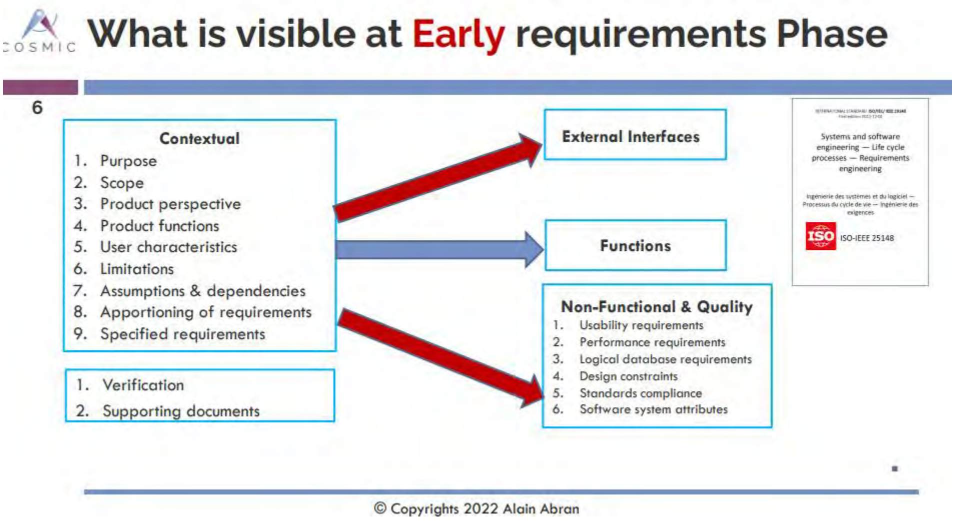
- Requirements do not describe the full scope of functionality of the software with all the necessary functional details.
  - Most of the time: requirements will be detailed & changed as the project moves through the life cycle.

© Copyrights 2022 Alain Abran



Key Concept: from Simple to Complex Software

5



What is visible at Early requirements Phase

6

Contextual

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

1. Verification
2. Supporting documents

External Interfaces

Functions

Non-Functional & Quality

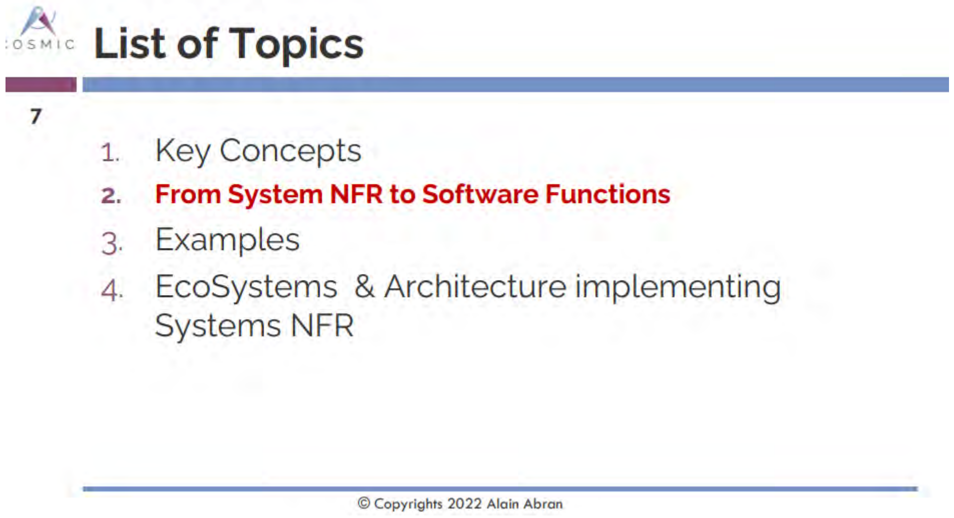
1. Usability requirements
2. Performance requirements
3. Logical database requirements
4. Design constraints
5. Standards compliance
6. Software system attributes

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

Systems and software engineering — Life cycle processes — Requirements engineering

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

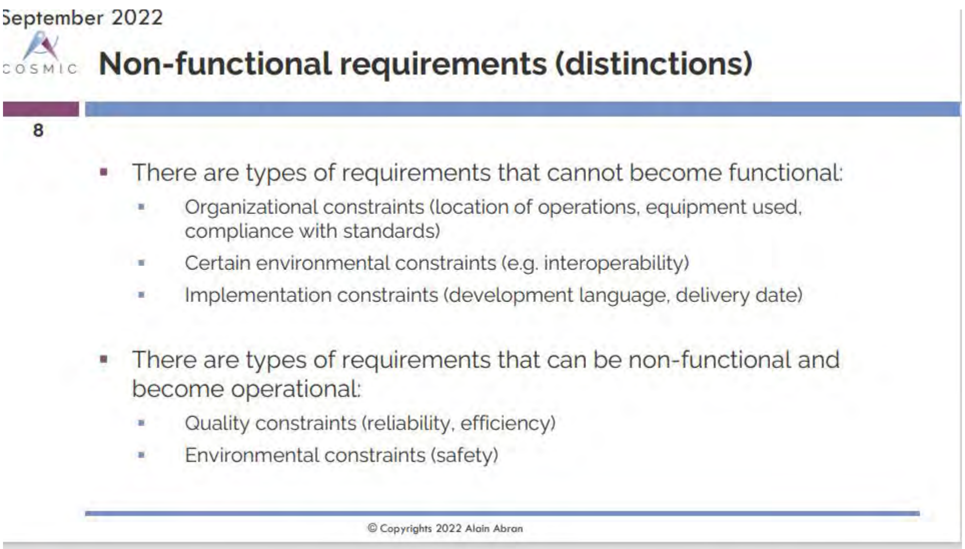
ISO ISO-IEEE 25148



List of Topics

7

1. Key Concepts
2. **From System NFR to Software Functions**
3. Examples
4. EcoSystems & Architecture implementing Systems NFR



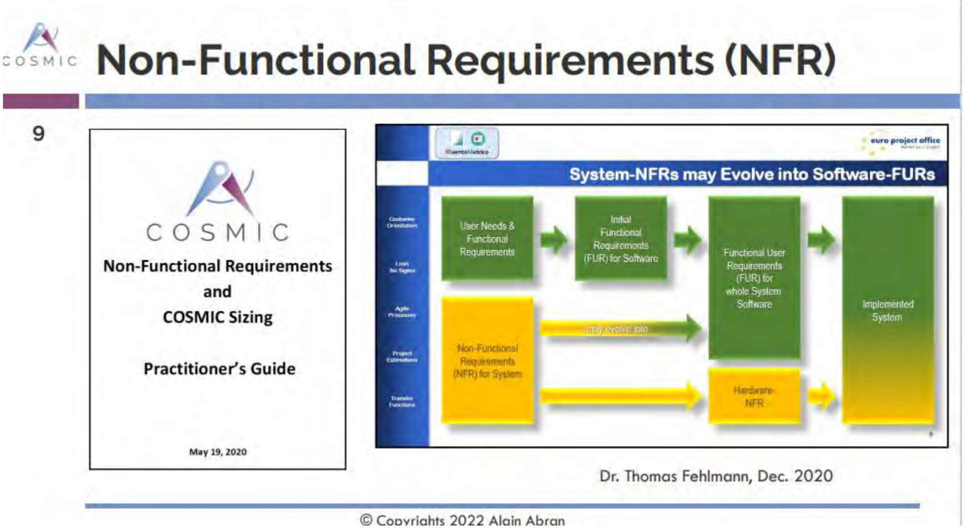
September 2022

Non-functional requirements (distinctions)

8

- There are types of requirements that cannot become functional:
  - Organizational constraints (location of operations, equipment used, compliance with standards)
  - Certain environmental constraints (e.g. interoperability)
  - Implementation constraints (development language, delivery date)
- There are types of requirements that can be non-functional and become operational:
  - Quality constraints (reliability, efficiency)
  - Environmental constraints (safety)

© Copyrights 2022 Alain Abran



Non-Functional Requirements (NFR)

9

COSMIC

Non-Functional Requirements and COSMIC Sizing

Practitioner's Guide

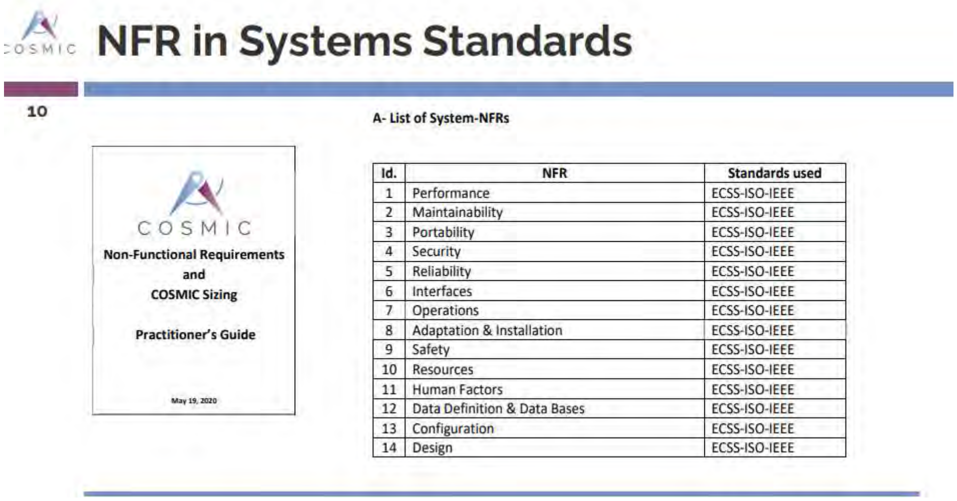
May 19, 2020

System-NFRs may Evolve into Software-FURs

euro project office
Customer Orientation
Lean Six Sigma
Agile Processes
Project Estimations
Transfer Functions
User Needs & Functional Requirements
Initial Functional Requirements (FUR) for Software
Functional User Requirements (FUR) for whole System Software
Implemented System
may evolve into
Non-Functional Requirements (NFR) for System
Hardware-NFR

Dr. Thomas Fehlmann, Dec. 2020

© Copyrights 2022 Alain Abran



NFR in Systems Standards

10

COSMIC

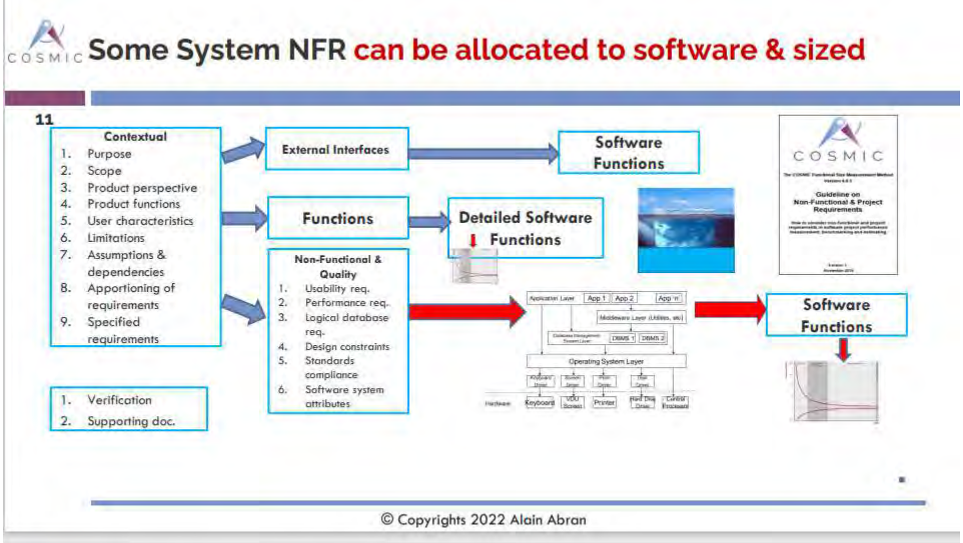
Non-Functional Requirements and COSMIC Sizing

Practitioner's Guide

May 19, 2020

A- List of System-NFRs

Id.	NFR	Standards used
1	Performance	ECSS-ISO-IEEE
2	Maintainability	ECSS-ISO-IEEE
3	Portability	ECSS-ISO-IEEE
4	Security	ECSS-ISO-IEEE
5	Reliability	ECSS-ISO-IEEE
6	Interfaces	ECSS-ISO-IEEE
7	Operations	ECSS-ISO-IEEE
8	Adaptation & Installation	ECSS-ISO-IEEE
9	Safety	ECSS-ISO-IEEE
10	Resources	ECSS-ISO-IEEE
11	Human Factors	ECSS-ISO-IEEE
12	Data Definition & Data Bases	ECSS-ISO-IEEE
13	Configuration	ECSS-ISO-IEEE
14	Design	ECSS-ISO-IEEE



Some System NFR can be allocated to software & sized

11

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

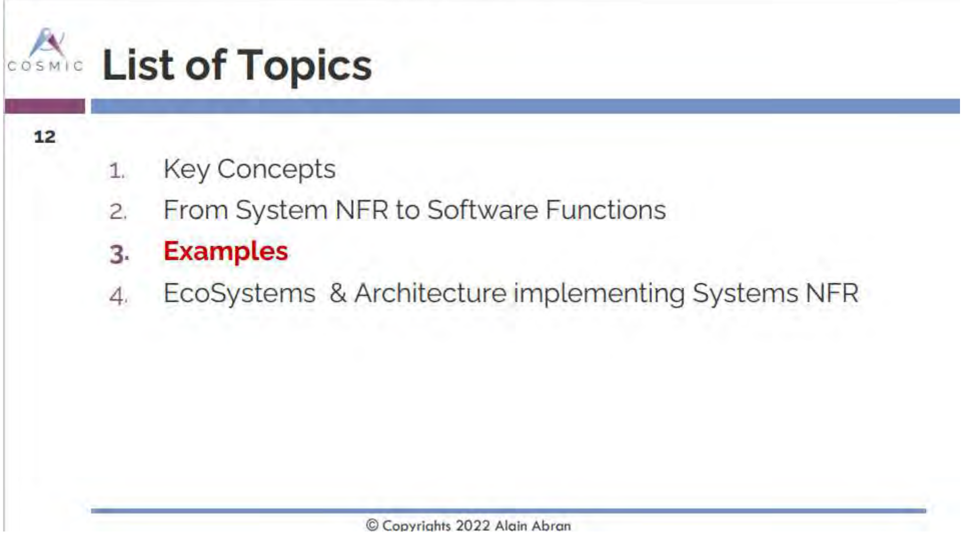
1. Verification
2. Supporting doc.

External Interfaces → Software Functions

Functions → Detailed Software Functions

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

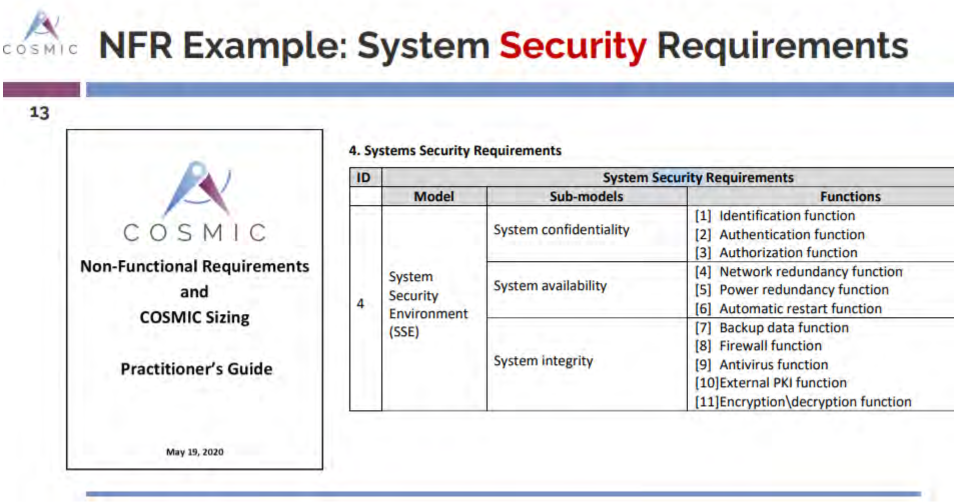
→ Software Functions



List of Topics

12

1. Key Concepts
2. From System NFR to Software Functions
3. **Examples**
4. EcoSystems & Architecture implementing Systems NFR



NFR Example: System Security Requirements

13

COSMIC

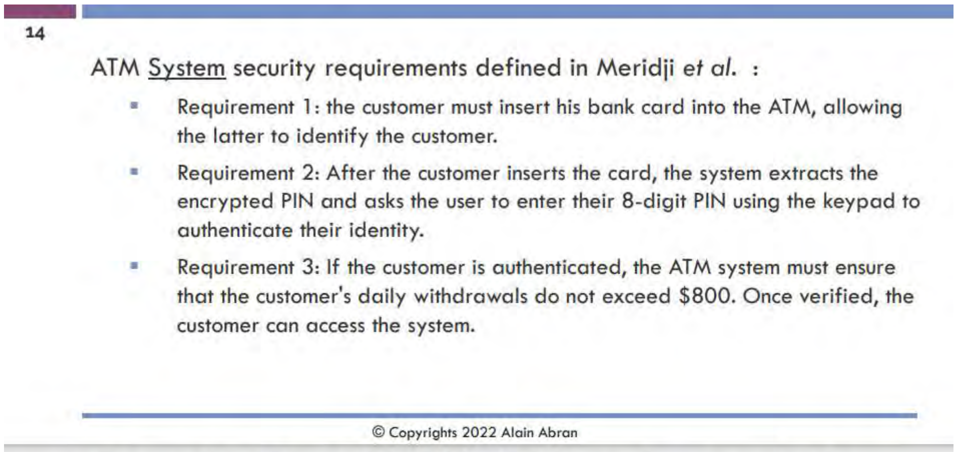
Non-Functional Requirements and COSMIC Sizing

Practitioner's Guide

May 19, 2020

4. Systems Security Requirements

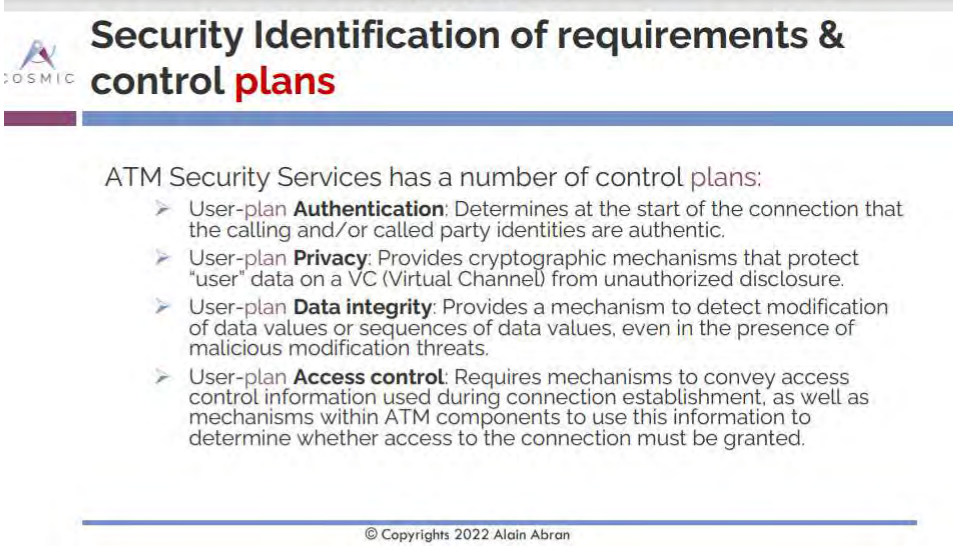
ID	System Security Requirements		
	Model	**Sub-models**	**Functions**
4	System Security Environment (SSE)	System confidentiality	[1] Identification function [2] Authentication function [3] Authorization function
		System availability	[4] Network redundancy function [5] Power redundancy function [6] Automatic restart function
		System integrity	[7] Backup data function [8] Firewall function [9] Antivirus function [10]External PKI function [11]Encryption\decryption function



14

ATM System security requirements defined in Meridji *et al.* :

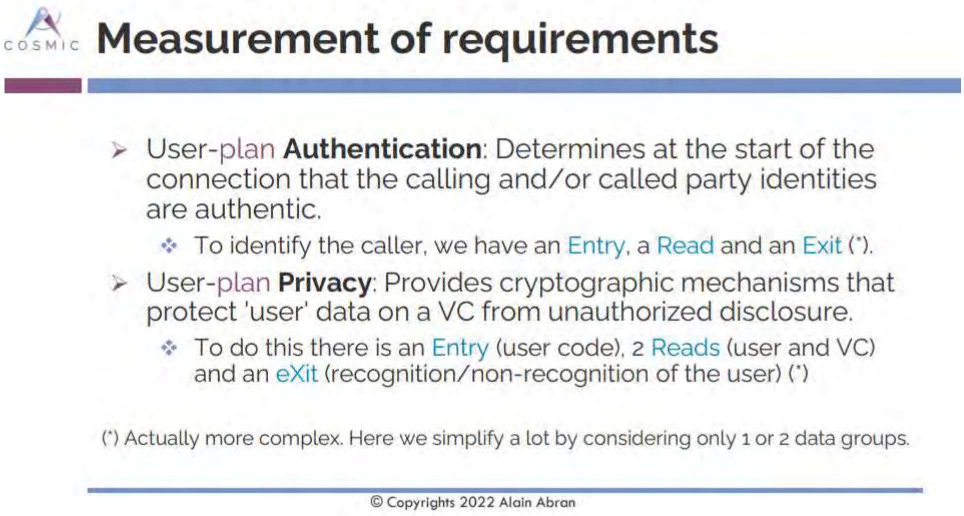
- Requirement 1: the customer must insert his bank card into the ATM, allowing the latter to identify the customer.
- Requirement 2: After the customer inserts the card, the system extracts the encrypted PIN and asks the user to enter their 8-digit PIN using the keypad to authenticate their identity.
- Requirement 3: If the customer is authenticated, the ATM system must ensure that the customer's daily withdrawals do not exceed $800. Once verified, the customer can access the system.



Security Identification of requirements & control plans

ATM Security Services has a number of control plans:

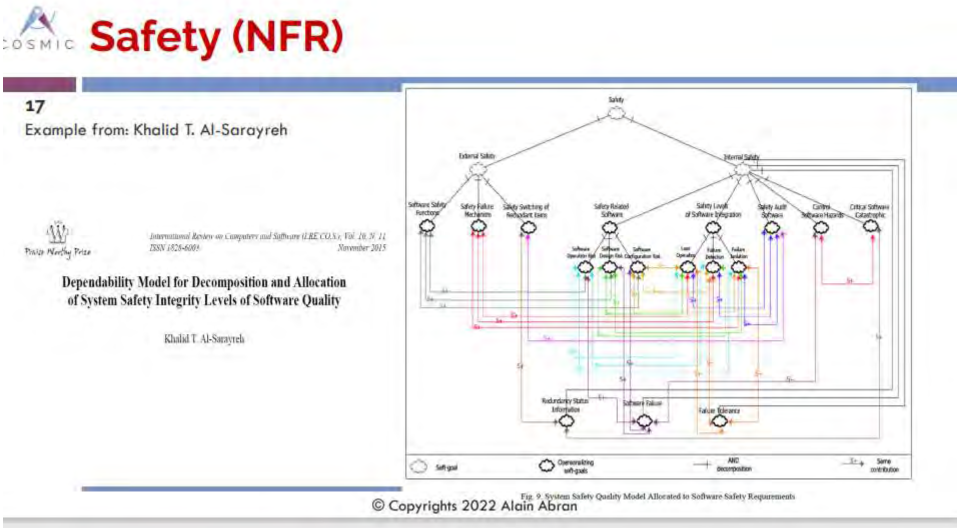
- User-plan **Authentication**: Determines at the start of the connection that the calling and/or called party identities are authentic.
- User-plan **Privacy**: Provides cryptographic mechanisms that protect "user" data on a VC (Virtual Channel) from unauthorized disclosure.
- User-plan **Data integrity**: Provides a mechanism to detect modification of data values or sequences of data values, even in the presence of malicious modification threats.
- User-plan **Access control**: Requires mechanisms to convey access control information used during connection establishment, as well as mechanisms within ATM components to use this information to determine whether access to the connection must be granted.



Measurement of requirements

- User-plan **Authentication**: Determines at the start of the connection that the calling and/or called party identities are authentic.
  - To identify the caller, we have an Entry, a Read and an Exit (*).
- User-plan **Privacy**: Provides cryptographic mechanisms that protect 'user' data on a VC from unauthorized disclosure.
  - To do this there is an Entry (user code), 2 Reads (user and VC) and an eXit (recognition/non-recognition of the user) (*)

(*) Actually more complex. Here we simplify a lot by considering only 1 or 2 data groups.



Safety (NFR)

17

Example from: Khalid T. Al-Sarayreh

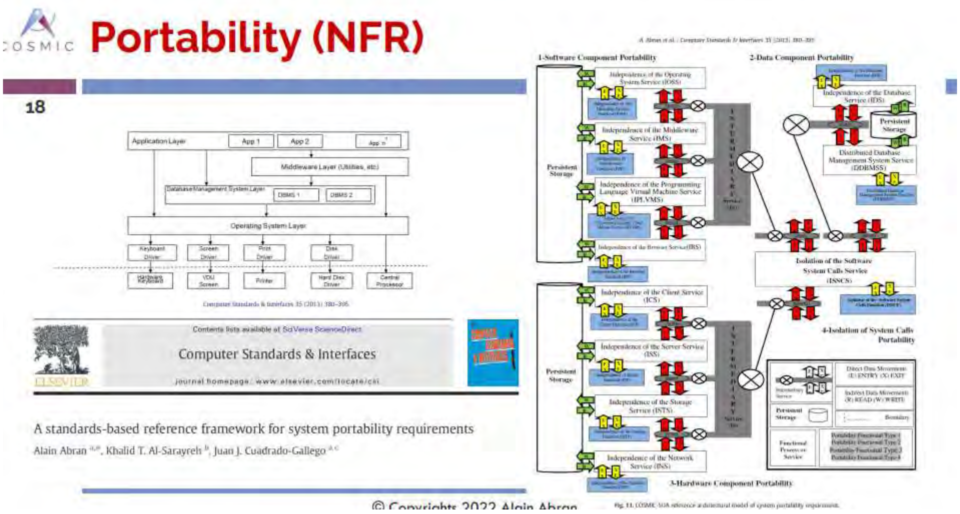
International Review on Computers and Software (I.RE.CO.S.), Vol. 10, N. 11
ISSN 1828-6003 November 2015

Dependability Model for Decomposition and Allocation of System Safety Integrity Levels of Software Quality

Khalid T. Al-Sarayreh

Fig. 9. System Safety Quality Model Allocated to Software Safety Requirements

© Copyrights 2022 Alain Abran



Portability (NFR)

18

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

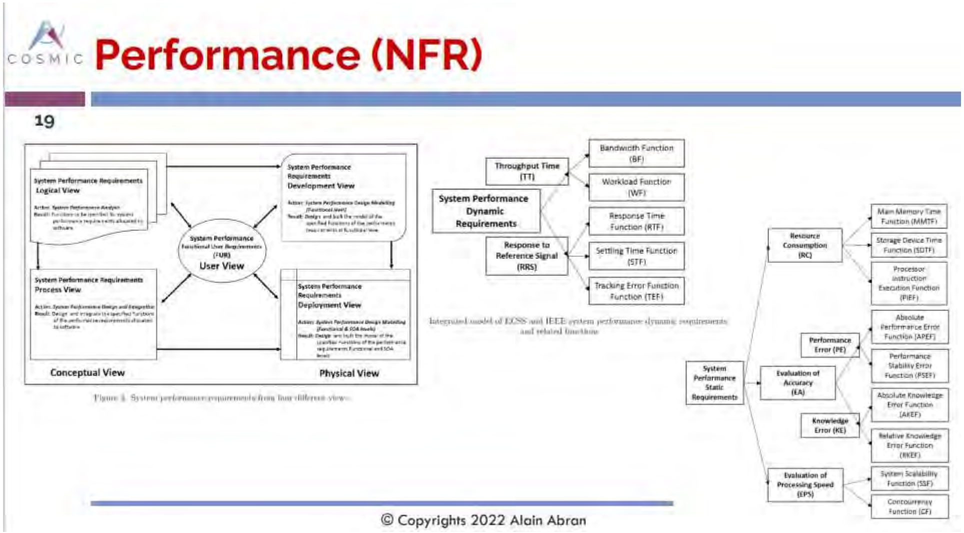
Alain Abran [a,*], Khalid T. Al-Sarayreh [b], Juan J. Cuadrado-Gallego [a,c]

A. Abran et al. / Computer Standards & Interfaces 35 (2013) 380–395

ELSEVIER
1-Software Component Portability
2-Data Component Portability
3-Hardware Component Portability
4-Isolation of System Calls Portability
Independence of the Operating System Service (IOSS)
Independence of the Middleware Service (IMS)
Independence of the Programming Language Virtual Machine Service (IPLVMS)
Independence of the Browser Service(IBS)
Independence of the Client Service (ICS)
Independence of the Server Service (ISS)
Independence of the Storage Service (ISTS)
Independence of the Network Service (INS)
Independence of the Database Service (IDS)
Distributed Database Management System Service (DDBMSS)
Isolation of the Software System Calls Service (ISSCS)
Persistent Storage
INTERMEDIARY Service
Direct Data Movements (E) ENTRY (X) EXIT
Indirect Data Movements (R) READ (W) WRITE
Boundary
Functional Process or Service
Portability Functional Type 1
Portability Functional Type 2
Portability Functional Type 3
Portability Functional Type 4

Fig. 11. COSMIC SOA reference architectural model of system portability requirements.

© Copyrights 2022 Alain Abran



Performance (NFR)

19

System Performance Requirements Logical View
Action: System Performance Analysis
Result: Functions to be specified for system performance requirements allocated to software
System Performance Requirements Development View
Action: System Performance Design Modelling (Functional level)
Result: Design and built the model of the specified functions of the performance requirements at functional view
System Performance Functional User Requirements (FUR)
User View
System Performance Requirements Process View
Action: System Performance Design and Integration
Result: Design and integrate the specified functions of the performance requirements allocated to software
System Performance Requirements Deployment View
Action: System Performance Design Modelling (Functional & SOA levels)
Result: Design and built the model of the specified functions of the performance requirements Functional and SOA levels
Conceptual View
Physical View

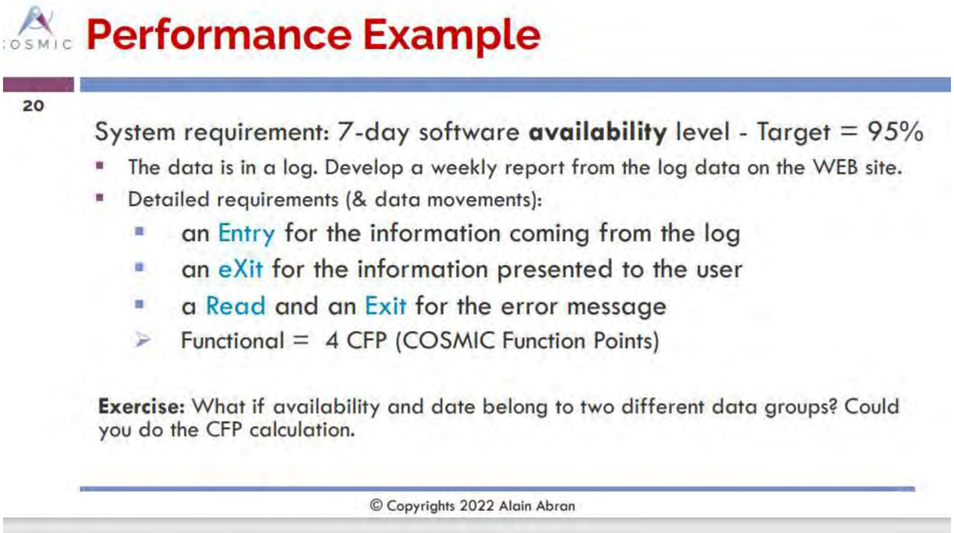
Figure 1. System performance requirements from four different views

System Performance Dynamic Requirements
Throughput Time (TT)
Response to Reference Signal (RRS)
Bandwidth Function (BF)
Workload Function (WF)
Response Time Function (RTF)
Settling Time Function (STF)
Tracking Error Function Function (TEF)

Integrated model of ECSS and IEEE system performance dynamic requirements and related functions

System Performance Static Requirements
Resource Consumption (RC)
Evaluation of Accuracy (EA)
Performance Error (PE)
Knowledge Error (KE)
Evaluation of Processing Speed (EPS)
Main Memory Time Function (MMTF)
Storage Device Time Function (SDTF)
Processor instruction Execution Function (PIEF)
Absolute Performance Error Function (APEF)
Performance Stability Error Function (PSEF)
Absolute Knowledge Error Function (AKEF)
Relative Knowledge Error Function (RKEF)
System Scalability Function (SSF)
Concourrency Function (CF)

© Copyrights 2022 Alain Abran



COSMIC

Performance Example

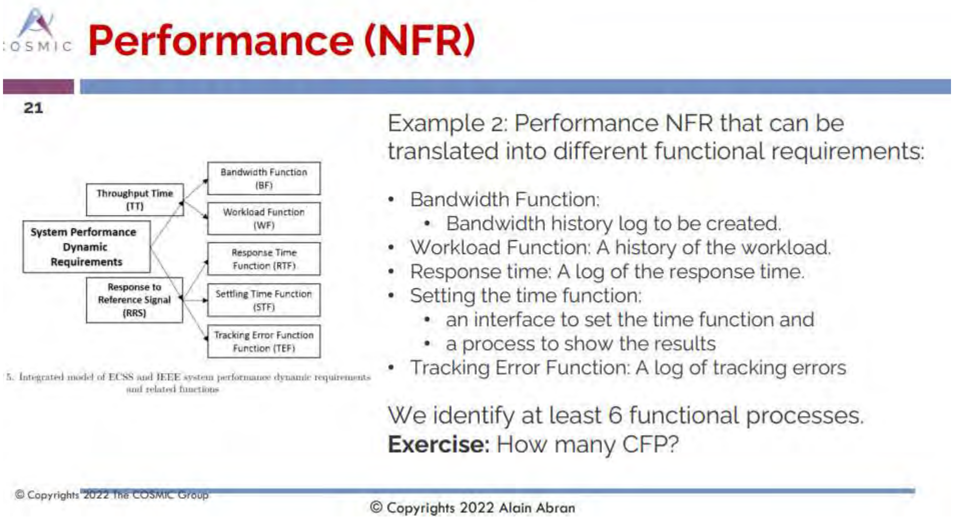
20

System requirement: 7-day software **availability** level - Target = 95%

- The data is in a log. Develop a weekly report from the log data on the WEB site.
- Detailed requirements (& data movements):
  - an Entry for the information coming from the log
  - an eXit for the information presented to the user
  - a Read and an Exit for the error message
  - ➢ Functional = 4 CFP (COSMIC Function Points)

Exercise: What if availability and date belong to two different data groups? Could you do the CFP calculation.

© Copyrights 2022 Alain Abran



COSMIC

Performance (NFR)

21

System Performance Dynamic Requirements
Throughput Time (TT)
Response to Reference Signal (RRS)
Bandwidth Function (BF)
Workload Function (WF)
Response Time Function (RTF)
Settling Time Function (STF)
Tracking Error Function Function (TEF)

5. Integrated model of ECSS and IEEE system performance dynamic requirements and related functions

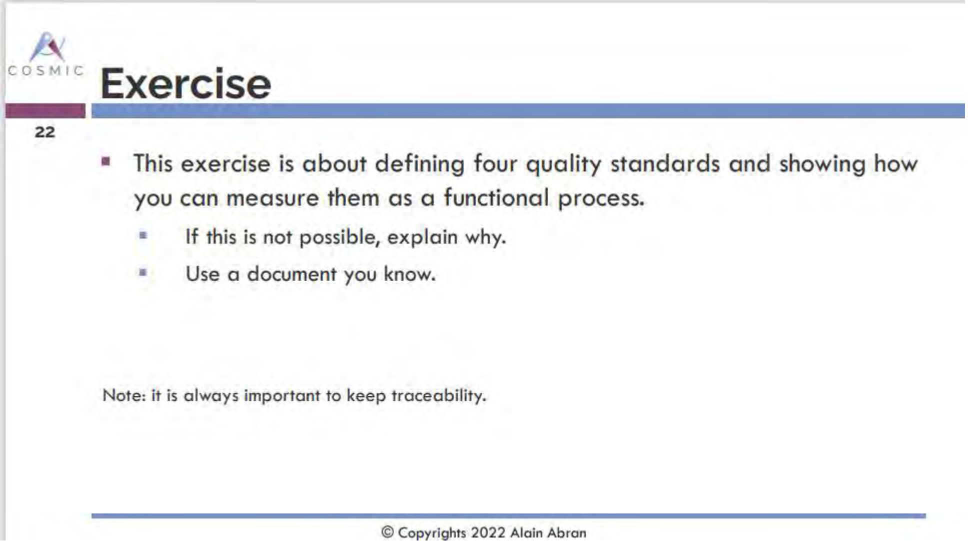
Example 2: Performance NFR that can be translated into different functional requirements:

- Bandwidth Function:
  - Bandwidth history log to be created.
- Workload Function: A history of the workload.
- Response time: A log of the response time.
- Setting the time function:
  - an interface to set the time function and
  - a process to show the results
- Tracking Error Function: A log of tracking errors

We identify at least 6 functional processes.
Exercise: How many CFP?

© Copyrights 2022 The COSMIC Group

© Copyrights 2022 Alain Abran



COSMIC

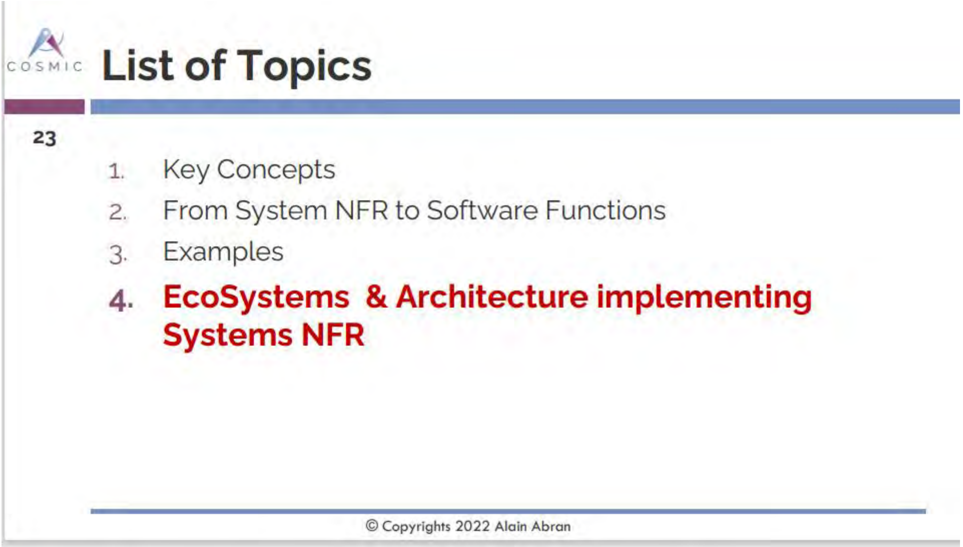
Exercise

22

- This exercise is about defining four quality standards and showing how you can measure them as a functional process.
  - If this is not possible, explain why.
  - Use a document you know.

Note: it is always important to keep traceability.

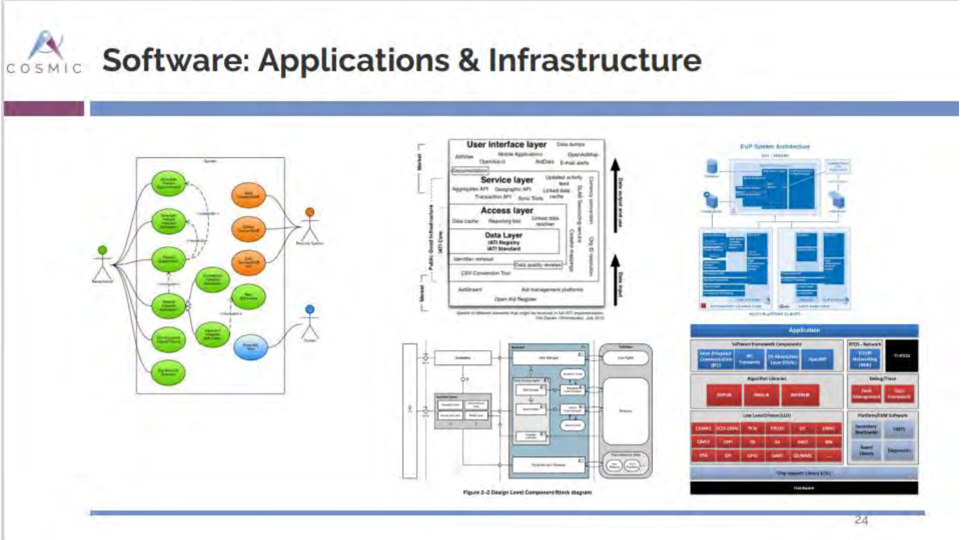
© Copyrights 2022 Alain Abran



List of Topics

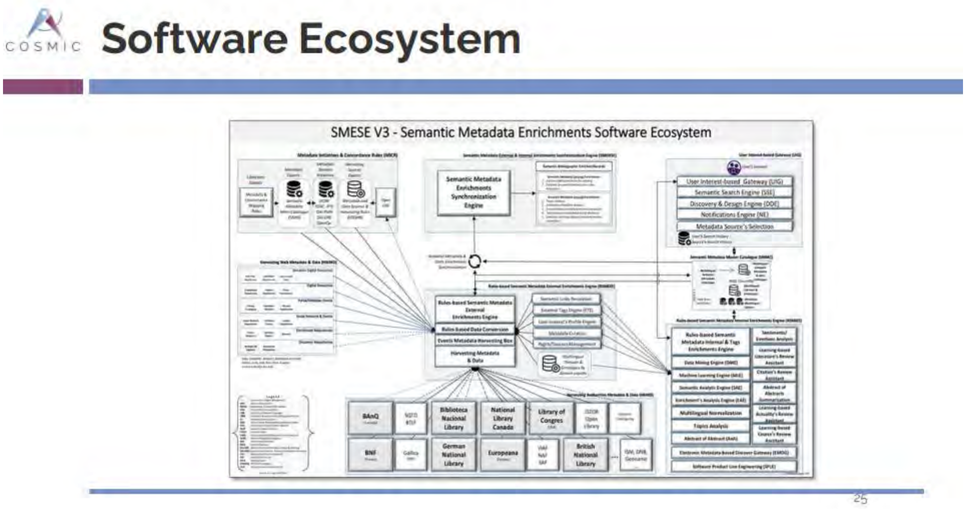
23

1. Key Concepts
2. From System NFR to Software Functions
3. Examples
4. **EcoSystems & Architecture implementing Systems NFR**



Software: Applications & Infrastructure

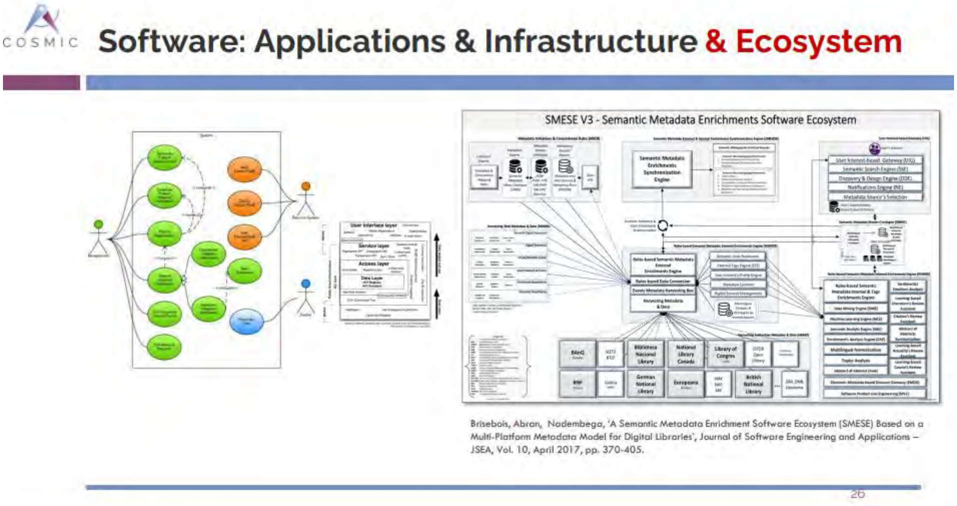
24



Software Ecosystem

SMESE V3 - Semantic Metadata Enrichments Software Ecosystem
Semantic Metadata Enrichments Synchronization Engine
User Interest-based Gateway (UIG)
Semantic Search Engine (SSE)
Discovery & Design Engine (DDE)
Notifications Engine (NE)
Metadata Source's Selection
Rules-based Semantic Metadata External Enrichments Engine
Rules-based Data Conversion
Events Metadata Harvesting Box
Harvesting Metadata & Data
BAnQ
BNF
Biblioteca Nacional Library
German National Library
National Library Canada
Europeana
Library of Congress
British National Library

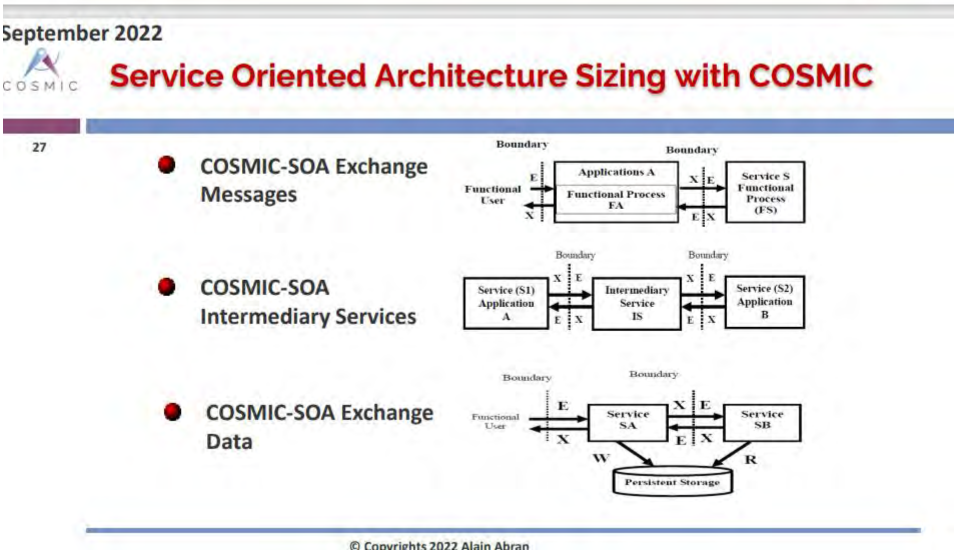
25



Software: Applications & Infrastructure & Ecosystem

Brisebois, Abran, Nadembega, 'A Semantic Metadata Enrichment Software Ecosystem (SMESE) Based on a Multi-Platform Metadata Model for Digital Libraries', Journal of Software Engineering and Applications – JSEA, Vol. 10, April 2017, pp. 370-405.

26



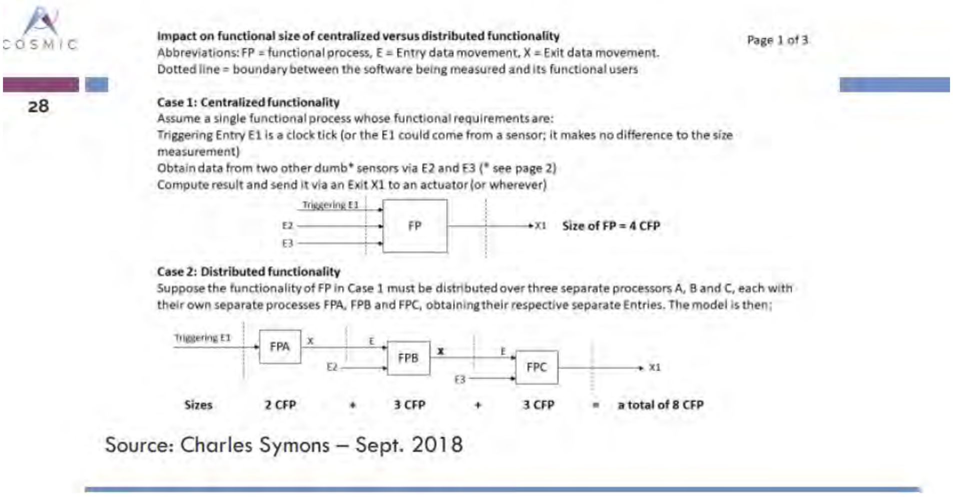
September 2022

Service Oriented Architecture Sizing with COSMIC

27

- COSMIC-SOA Exchange Messages
- COSMIC-SOA Intermediary Services
- COSMIC-SOA Exchange Data

© Copyrights 2022 Alain Abran



28

Impact on functional size of centralized versus distributed functionality

Page 1 of 3

Abbreviations: FP = functional process, E = Entry data movement, X = Exit data movement.
Dotted line = boundary between the software being measured and its functional users

Case 1: Centralized functionality

Assume a single functional process whose functional requirements are:
Triggering Entry E1 is a clock tick (or the E1 could come from a sensor; it makes no difference to the size measurement)
Obtain data from two other dumb* sensors via E2 and E3 (* see page 2)
Compute result and send it via an Exit X1 to an actuator (or wherever)

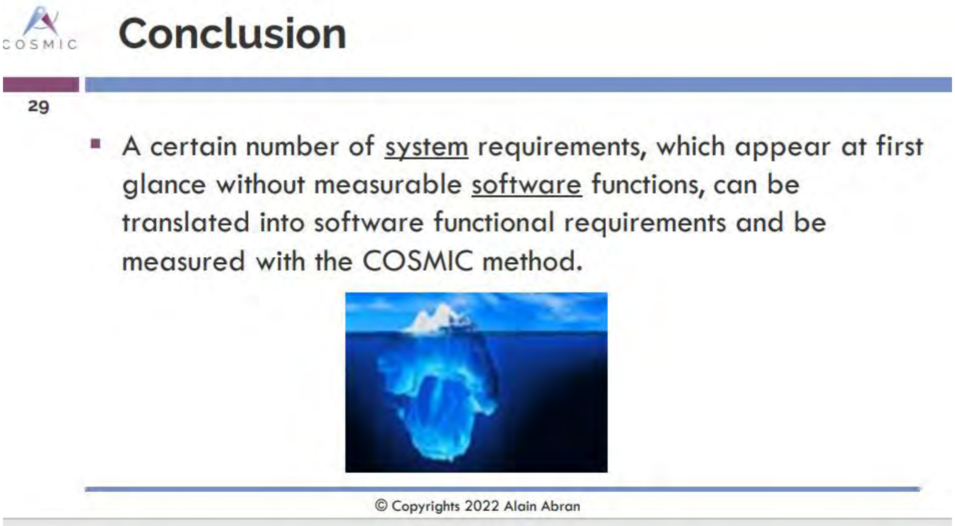
Size of FP = 4 CFP

Case 2: Distributed functionality

Suppose the functionality of FP in Case 1 must be distributed over three separate processors A, B and C, each with their own separate processes FPA, FPB and FPC, obtaining their respective separate Entries. The model is then;

Sizes 2 CFP + 3 CFP + 3 CFP = a total of 8 CFP

Source: Charles Symons – Sept. 2018

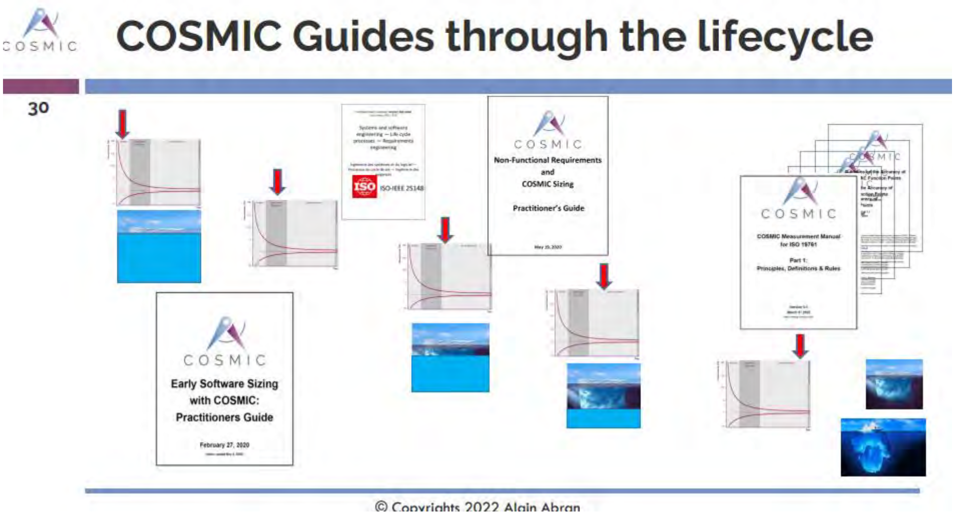


Conclusion

29

- A certain number of system requirements, which appear at first glance without measurable software functions, can be translated into software functional requirements and be measured with the COSMIC method.

© Copyrights 2022 Alain Abran



COSMIC Guides through the lifecycle

30

Systems and software engineering — Life cycle processes — Requirements engineering

ISO-IEEE 25148

COSMIC

Non-Functional Requirements and COSMIC Sizing

Practitioner's Guide

May 25, 2020

COSMIC

Early Software Sizing with COSMIC: Practitioners Guide

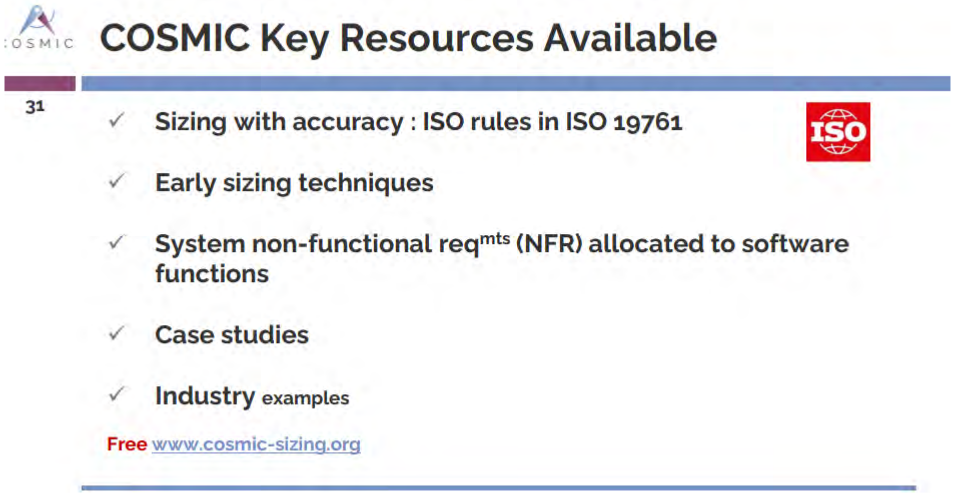
February 27, 2020

COSMIC

COSMIC Measurement Manual for ISO 19761

Part 1: Principles, Definitions & Rules

© Copyrights 2022 Alain Abran

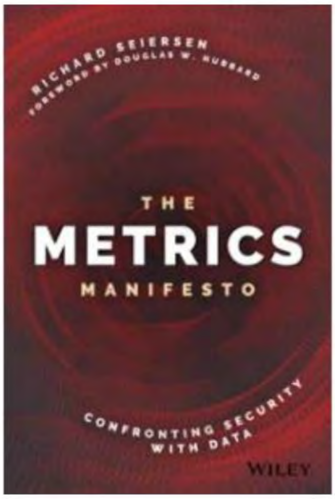


COSMIC Key Resources Available

31

- ✓ Sizing with accuracy : ISO rules in ISO 19761
- ✓ Early sizing techniques
- ✓ System non-functional req^{mts} (NFR) allocated to software functions
- ✓ Case studies
- ✓ Industry examples

Free www.cosmic-sizing.org

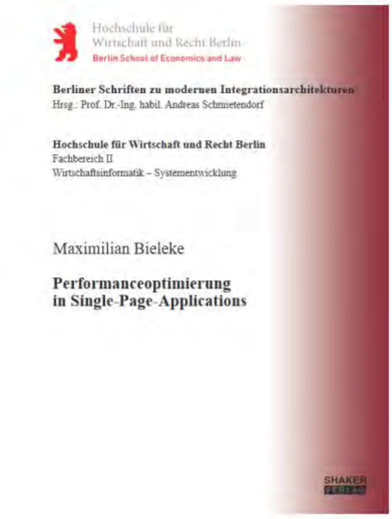


Seiersen, R.:

The Metrics Manifesto: Confronting Security with Data

John Wiley Publ., 2022, ISBN 978-1-119-51536-4

The Metrics Manifesto considers security with data delivers an examination of security metrics with R, the popular open source programming language and software development environment for statistical computing. This insightful and up to-date guide offers readers a practical focus on applied measurement that can prove or disprove the efficacy of information security measures taken by a firm. The book's detailed chapters combine topics like security, predictive analytics, and R programming to present an authoritative and innovative approach to security metrics

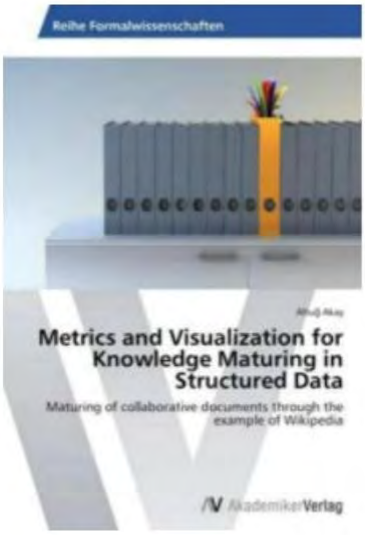

Hochschule für
Wirtschaft und Recht Berlin
Berlin School of Economics and Law
Berliner Schriften zu modernen Integrationsarchitekturen
Hrsg.: Prof. Dr.-Ing. habil. Andreas Schmietendorf
Hochschule für Wirtschaft und Recht Berlin
Fachbereich II
Wirtschaftsinformatik – Systementwicklung
Maximilian Bieleke
Performanceoptimierung
in Single-Page-Applications
SHAKER
VERLAG

Maxemilian Bieleke:

Performanceoptimierung in Single-Page Application

Shaker-Verlag, Aachen, 2021, ISBN 978-3-8440-8315-6

Das vorliegende Buch beschreibt die Effizienz von Web-Applikationen hinsichtlich deren Performance in ausgewählten Anwendungsbereichen.


Reihe Formalwissenschaften
Alfudj Akay
Metrics and Visualization for
Knowledge Maturing in
Structured Data
Maturing of collaborative documents through the
example of Wikipedia
AkademikerVerlag

Akay, A.:

Metrics and Visualization for Knowledge Maturing in Structured Data

Akademiker-Verlag, 2021

This book considers the maturing of information in collaborative environments such as Wikis, intranet documents or documents in cloud is investigated via four metrics. After the definition and calculation of the metrics, the results are visualized in graphical format. Therefore, the readers can see the evolution of the metrics within the time, but also the relations of metrics with each other.

Andreas Schmietendorf

ESAPI 2020 – 4. Workshop Evaluation of Service-APIs

Shaker-Verlag, Aachen, November 2020, ISBN 978-3-8440-7515-1

Das vorliegende Buch fasst die insgesamt 11 Beiträge und Diskussionen des 4. Workshop zur Bewertung von service-basierten APIs zusammen und ist in der Buchreihe der Schriften zu modernen Integrationsarchitekturen erschienen.

Sandro Hartenstein, Konrad Nadobny, Steven Schmidt, Andreas Schmietendorf
Sicherheits- und Compliance-Management im Lebenszyklus von Web-APIs
Secure Web-APIs
λογος

Hartenstein/Nadobny/Schmidt/ Schmietendorf:

Sicherheits- und Compliance Management

Logos-Verlag, Berlin, 2020
ISBN 978-3-8525-5086-8

This book describes approaches and techniques for implementing Web APIs keeping security-related requirements. The API management involves analytical and constructive approaches for quality assurance during the development. The DevOps approach was considered in the context of business processes.

Software Metrics
A Complete Guide - 2021 Edition
PRACTICAL TOOLS FOR SELF-ASSESSMENT

Software Metrics: A Complete Guide - 2021 Edition

Gerardus Blokdyk and Publishers, 2021
ISBN 978-1-8674-9201-6

This book summarizes essential software project and management metrics and their application to practical and industrial areas and examples.

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Berliner Schriften zu modernen Integrationsarchitekturen
Hrsg.: Prof. Dr.-Ing. habil. Andreas Schmietendorf

Hochschule für Wirtschaft und Recht Berlin
Fachbereich II
Wirtschaftsinformatik – Systementwicklung

Enterprise Computing Conference 2020

Ausgewählte Ergebnisse der Podiumsdiskussion, Diskussionsrunden und Workshops

12. bis 13. März 2020, Köln

SHAKER VERLAG

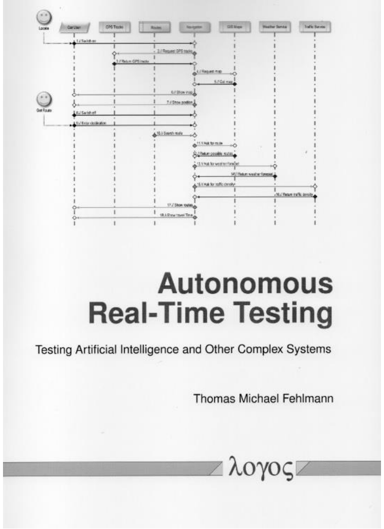
Schmietendorf, A.:

Enterprise Computing Conference 2020

Köln, März 2020

Shaker Verlag, Aachen, 2020,
ISBN 978-3-8440-7320-1

Dieses Buch beinhaltet die Beiträge zur ECC-Konferenz 2020 zur Thematik „Enterprise Transformation“ vor allem in relevanten Anwendungsbereichen.



Autonomous Real-Time Testing

Testing Artificial Intelligence and Other Complex Systems

Thomas Michael Fehlmann

λογος

Thomas M. Fehlmann:

Autonomous Real-Time Testing

Testing Artificial Intelligence and Other Complex Systems
Logos-Verlag, Berlin, 2020
ISBN 978-3-8525-5086-8

The book explains the theory and the implementation approach for a framework for Autonomous Real-Time Testing (ART) of a software-intense system while in operation. Principles and approaches like Combinatory logic,Analytic Hierarchy Process (AHP) and Quality Function Deployment (QFD) are used for a complex testing approach of real-time systems like automotive solutions, IoT control software and embedded system releases.

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Berliner Schriften zu modernen Integrationsarchitekturen
Hrsg.: Prof. Dr.-Ing. habil. Andreas Schmietendorf

Hochschule für Wirtschaft und Recht Berlin
Fachbereich II
Wirtschaftsinformatik – Systementwicklung

Empirische Untersuchungen zum Cloud-Einsatz im KMU-Bereich - eine zusammenfassende Betrachtung

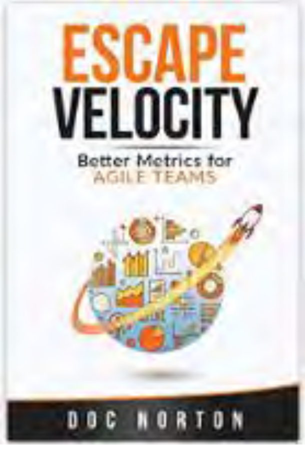
SHAKER VERLAG

Andreas Schmietendorf

Empirische Untersuchungen zum Cloud-Einsatz im KMU-Bereich - eine zusammenfassende Betrachtung

Shaker-Verlag, Aachen, April 2020, ISBN 978-3-8440-7356-0

Das vorliegende Buch reflektiert die Ergebnisse von forschungs- aber auch industrieorientierten Projekten rund um die Themenstellung des Cloud Computings, die durch den Autor initiiert und in den vergangenen 10 Jahren verantwortet bzw. im Rahmen seiner Arbeitsgruppe bearbeitet wurden.



Doc Norton

Escape Velocity: Better Mertrics for Agile Teams

Februar 2020

This book identifies the velocity as the most commonly used metric in agile software delivery. The efficiency of Scrum teams is the main focus. Metrics are considered in general and further measure like lead time, team joy, team performance etc. are proposed especially. The book includes many interesting stories of agile team management.

Elyjoy Muthoni Micheni

Metrtics and Models for Evaluating the Quality and Effectiveness of ERP Software

Juli 2019

This book presents a set of theoretical measurement models and metrics for measuring software size and complexity of large scale enterprise resource planning software based on practical experiences. It focuses on the measurement of usability, service quality, security, interoperability, maintenance and enterprise resource planning.

Agile Project Management with Azure DevOps
Concepts, Templates, and Metrics
Joachim Rossberg
Apress

Joachim Rosberg

Agile Project Management with Azure DevOps: Concepts, Templates, and Metrics

April 2019

This book considers Agile project management to use and customize Microsoft Azure DevOps. The basic process involves the Application Life Cycle Management approach and achieve an overall higher quality output.“

Hochschule für Wirtschaft und Recht Berlin
Berlin School of Economics and Law
Berliner Schriften zu modernen Integrationsarchitekturen
Hrsg.: Prof. Dr.-Ing. habil. Andreas Schmietendorf
Hochschule für Wirtschaft und Recht Berlin
Fachbereich II
Wirtschaftsinformatik · Systementwicklung
ESAPI 2019
3. Workshop Evaluation of Service APIs
07. November 2019, Dresden

Schmietendorf, A.:

Workshop ESAPI 2019

Dresden, November 2019

Shaker Verlag, Aachen, 2019,
ISBN 978-3-8440-6837-5

Dieses Buch beinhaltet die Beiträge zur ESAPI-Konferenz 2019 zu Sicherheits- und Compliance-aspekten von Web-APIs vor allem in relevanten Anwendungsbereichen.

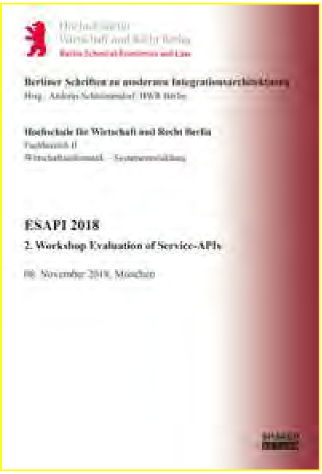
Miroslaw Staron
Wilhelm Meding
Software Development Measurement Programs
Development, Management and Evolution
Springer

Miroslaw Staron:

Software Development Measurement Programs

Springer Publ., 2019
ISBN 978-3030063085

This book describes approaches and techniques for implementing software measurement processes from a practical point of view involving toll support, project integration and measurement programs evolution.



Berliner Schriften zu modernen Integrationsarchitekturen
Hrsg.: Andreas Schmietendorf, HWR Berlin

Hochschule für Wirtschaft und Recht Berlin
Fachbereich II
Wirtschaftsinformatik – Systementwicklung

ESAPI 2018
2. Workshop Evaluation of Service-APIs
08. November 2018, München

Schmietendorf, A.,:

ESAPI 2018

2. Workshop: Evaluation of Service-APIs
8. November 2018, München

Shaker Verlag, Aachen, April 2018, ISBN 978-3-8440-6254-0

The book includes the proceedings of the Evaluation of Service-APIs 2018 Workshop held in Munich in November 2018, which constitute a collection of theoretical studies in the field of measurement and evaluation of service oriented and API technologies.

Software measurement
The Ultimate Step-By-Step Guide
PRACTICAL TOOLS FOR SELF-ASSESSMENT
The Art of Service

Gerardus Blokdyk:

Software Measurement the Ultimate Step-By-Step Guide

5starcooks Publ. 2018

This book summarizes some helpful practical experiences about measurement integration in software management processes and their successful implementation.

Berliner Schriften zu modernen Integrationsarchitekturen
Hrsg.: Andreas Schmietendorf, HWR Berlin
André Nitze, Ultra Tendency GmbH

Hochschule für Wirtschaft und Recht Berlin
Fachbereich II
Wirtschaftsinformatik – Systementwicklung

API-First/API-Management -
Open APIs als Treiber der Digitalisierung

Workshop im Rahmen der
Enterprise Computing Conference

19. April 2018, Hamburg

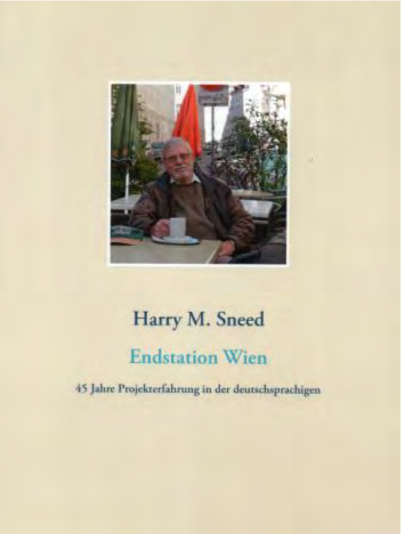
Schmietendorf, A., Nitze, A.:

ESAPI 2018

2. Workshop: API-First/API-Management
19. April, Hamburg

Shaker Verlag, Aachen, April 2018, ISBN 978-3-8440-5927-4

The book includes the proceedings of the API-First/API-Management 2018 Workshop held in Hamburg in April 2018, which constitute a collection of theoretical studies in the field of measurement and evaluation of service oriented and API technologies.



Harry Sneed:

Endstation Wien

45 Jahre Projekterfahrungen in der deutschsprachigen IT-Welt

BoD Norderstedt, 2017, 328 S.

ISBN 978-3-7448-8364-1

Dieses Buch beschreibt nahezu die gesamte Tätigkeit von Harry Sneed in der IT-Welt, von den Anfängen der Großrechner mit den COBOL und PL/1-Programmen bis hin zu den aktuellen und modernen Ansätzen Service-orientierter Technologien und Systemen. Dieses Buch fasst vor allem die umfangreichen Erfahrungen zu Wartungs-, Migrations- und Testprojekten zusammen, die auch für die Beherrschung aktueller und moderner Software-Anwendungen, von unschätzbarem Wert sind.

IWSM-Mensura 2017: Gothenburg, Sweden

> Home > Conferences and Workshops > IWSM-Mensura

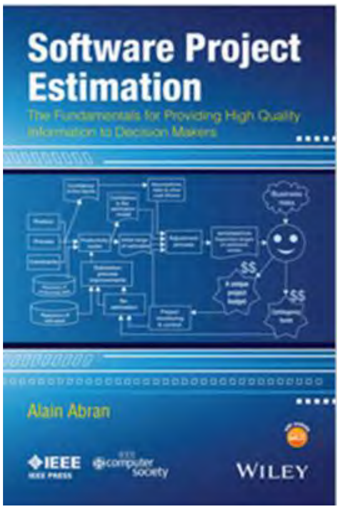
Miroslaw Staron, Wilhelm Meding:
Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement, IWSM-Mensura 2017, Gothenburg, Sweden, October 25 - 27, 2017. ACM 2017, ISBN 978-1-4503-4853-9

Staron, M, Melding, W.:

Proceedings of the IWSM/Mensura 2017

Joined Conference of the 27th International Workshop on Software Measurement (IWSM) and the 12th International Conference on Software Process and Product Measurement (Mensura), ACM 2017, *ISBN 978-1-4503-4853-9*

This proceedings are available at the Computer Science Bibliography of Trier.


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

Abran, A.:

Software Project Estimation: The Fundamentals for Providing

High Quality Information to Decision Makers

Wiley IEEE Computer Society Press, 2015 (288 pages), ISBN 978-1-118-95408-9

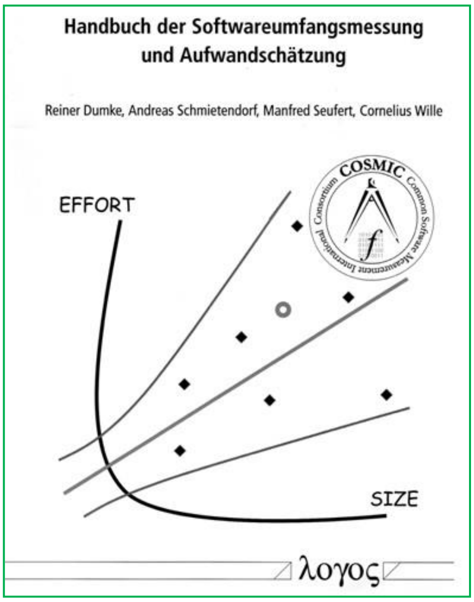
This book introduces theoretical concepts to explain the fundamentals of the design and evaluation of software estimation models. It provides software professionals with vital information on the best software management software out there. End-of-chapter exercises, Over 100 figures illustrating the concepts presented throughout the book, Examples incorporated with industry data.

Please remember:

Dumke, R., Schmietendorf, A., Seufert, M., Wille, C.:

Handbuch der Softwareumfangsmessung und Aufwandschätzung

Logos Verlag, Berlin, 2014 (570 Seiten), ISBN 978-3-8325-3784-5


Handbuch der Softwareumfangsmessung
und Aufwandschätzung
Reiner Dumke, Andreas Schmietendorf, Manfred Seufert, Cornelius Wille
EFFORT
SIZE
COSMIC
λογος

This book shows an overview about the current software size measurement and estimation approaches and methods. The essential part in this book gives a complete description of the ***COSMIC measurement method****, their application for different systems like embedded and business software and their use for cost and effort estimation based on this modern ISO size measurement standard.*

Software Measurement & Data Analysis Addressed Conferences

September 2022

Euromicro DSD/SEAA 2022:
Software Engineering & Advanced Application Conf.
August 31-Sept. 2,2022,Gran Canaria, Spain
see: https://dsd-seaa2022.iuma.ulpgc.es

EuroAsiaSPI2 2022:
European Systems & Software Process Improvement and Innovation Conference
August 31 - September 2, 2022, Salzburg, Austria
see: https://nqa2.iscn.com/index.php

OSS 2022:
International Conference on Open Source Systems
September 6 - 10, 2022, Madrid, Spain
see: https:/oss2022.github.io

QEST 2022:
International Conference on Quantitative Evaluation of SysTems
September 12 - 16, 2022, Warsaw, Poland
see: http://www.qest.org/qest2022/

ICPE 2022:
ACM/SPEC International Conference on Performance Engineering
September 15 – 16, 2022, Amsterdam, Netherlands
see: https://waset.org/performance-engineering-conference-in-september-2022-in-amsterdam

ESEM 2022:
Conference on Empirical Software Engineering and Measurement
September 19 - 23, 2022, Helsinki, Finland
see: https://conf.researchr.org/home/esem-2022

data2day 2022:
Konferenz für Big Data, Data Science und Machine Learning
September 20 - 21, 2022, Karlsruhe, Germany
see: https://www.data2day.de

SEFM 2022: International Conference on Software Engineering and Formal Methods
September 28 – 30, 2022, Berlin, Germany
see: https://sefm-conference.github.io/2022/

IWSM/Mensura 2022: Common International Conference on Software Measurement
28 – 31 September, Izmir, Turkey
see: https://www.iwsm-mensura.org/

October 2022

ASQT 2022: Arbeitskonferenz Softwarequalität, Test und Innovation
October , 2022, organization in process
see: http://www.asqt.org/

ASE 2022: Automated Software Engineering
October 10 - 14, 2022, Ann Arbor, Michigan, USA
see: https://conf.researchr.org/home/ase-2022

ICSEA 2022: International Conference on Software Engineering Advances
October 28 - 30, 2022, Can Tho, Vietnam
see: https:/icsea.org/index.html

November 2022

ESAPI 2022: API Conference 2022,
November 15 , 2022, Kaiserslautern, Germany
see: https://blog.hwr-berlin.de/schmietendorf/

ICPCC 2022: Performance Computing and Communications Conference
November 15 – 16, 2022, Jeddah, Saudi Arabia
see: https://waset.org/performance-computing-and-communications_conference-in-november-2022-in-jeddah

PROFES 2022:	International Conference on Product Focused Software Process Improvement November , 2022, Jyvaskyla, Finland see https://www.profes-conferences.org/
ESEC/FSE 2022:	European Software Engineering Conference and Symposium on the Foundation of Software Engineering November 14 - 18, 2022, Singapore see: https://2022.esec-fse.org/

December 2022

IEEE ICDM 2022:	IEEE International Conference on Data Mining November 30 - December 3 , 2022, Orlando, USA see: https://icdm22.cse.usf.edu/
ICDMCC 2022:	International Conference on Data Mining and Cloud Computing December 13 – 14, 2022, Cairo, Egypt see: https://waset.org/data-mining-and-cloud-computing-conference-in-december-2022-in-cairo
Big Data 2022:	IEEE International Conference on Big Data December 10 - 14, 2022, Orlando, USA see: http://www.servicessociety.org/bigdata

see also:

- http://www.acisinternational.org/newconferences.html
- https://www.acm.org/conferences
- https://www.ieee.org/conferences_events/index.html

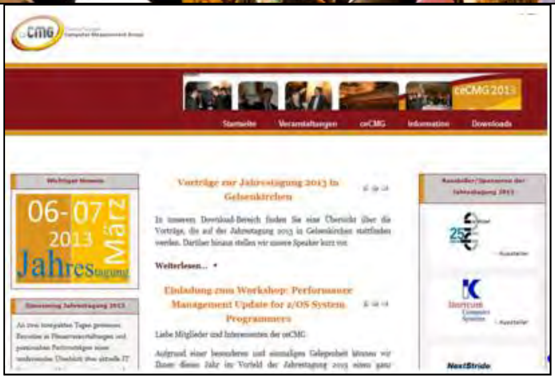
COMMUNITIES

A community of professionals

COSMIC started in 1998 and grew informally to a global community of thousands of professionals

Common Software Measurement International Consortium (COSMIC)

http://cosmic-sizing.org


ceCMG
ceCMG 2013
Startseite Veranstaltungen ceCMG Information Downloads
06-07 2013 März Jahrestagung
NextStride

Vorträge zur Jahrestagung 2013 in Gelsenkirchen

Einladung zum Workshop: Performance Management Update for z/OS System Programmers

Liebe Mitglieder und Interessenten der ceCMG

Central Europe Computer Measurement Group (ceCMG)

http://www.cecmg.de

MAIN
Metrics Associations' International Network
Home Associations Organisations Conferences and Events Projects
Member Logon Username Password Logon
Spain - AEMES

MAIN is an international network of autonomous software metrics associations.

Presentation by Manfred Bundschuh

Aims	Objectives
Run conferences in cooperation with national software metrics associations. Initiate and control common projects and working groups. Provide via this website: News, Announcements and Information Articles, Downloads, Discussion Forums, Webinars, Links to other Resources.	• Exchange of experience between organisations, individuals, and academia. • Promotion of measurement usage. • Harmonization of measurement issues, e.g. how to measure in certain environments, develop certified measurement expert curriculum. • Develop a common knowledgebase of documents such as metrics papers, case studies, training materials as standards, research initiatives, benchmark database, collection of best practices

News from MAIN Associations

Metrics Association's International Network (MAIN)

http://*www.mai-net.org*

Estimating

It is not about the size. Estimating is the translation of size to the things that really matter: time and money.

Your starting point for software measurement and software

Netherlands Software Metrics users Association (NESMA)

http://www.nesma.org/

Willkommen bei der Fachgruppe Measurement & Data Science

Die Fachgruppe "Measurement & Data Science" (FG 2.1.10) der Gesellschaft für Informatik e.V. ist das Kompetenzzentrum für Messung, Analyse und Bewertung von Software im IT-Bereich.

Inhaltlich befasst sich die Fachgruppe mit der Quantifizierung in der Softwaretechnik, also Bewertung und Metriken für Software-Systeme, Schätzung und Projektsteuerung, Kennzahlen für Performanz und Qualität, Risikomanagement, Messtheorie, Qualitätsmanagement, Data Analytics sowie empirisches Software Engineering.

Die Fachgruppe "Measurement & Data Science" bringt Experten aus der Forschung und Industrie zusammen und hat die folgenden Ziele:

- Plattform für Benchmarks, Netzwerke sowie Austausch zwischen Unternehmen,
- Bindeglied für Technologietransfer zwischen Forschung und Industrie

Aktuelles

Workshop: Vertrauenswürdigkeit von KI-Lösungen / Call for Paper

Wann: 15. November 2022 Wo: Fraunhofer IESE, Kaiserslautern (vor Ort oder virtuell) Veranstalter: GI-Fachgruppe "Measurement & Data Science" (FG...

weiterlesen

GI-Fachgruppe Measurement und Data Science

https://fg-data-science.gi.de/

(Measurement News Online)

Deutschsprachige Anwendergruppe für Software-Metrik und Aufwandschätzung e.V.

>> Welcome to DASMA ! <<

AKTUELLES

Umfrage zum Thema Cloud-Nutzung im Mittelstand

DASMA Nachwuchs

DASMA - Ihre erste Adresse rund um Software-Metriken und Aufwandschätzung

Die gemeinnützige DASMA e.V. verfolgt das Ziel, das Wissen und den Erfahrungsaustausch über die Messung und quantitative Bewertung von Software und ihrer Entwicklungsprozesse zu fördern und zu verbreiten. Diese unter dem Begriff Software-Metrik bekannten Bewertungsansätze

Deutschsprachige Anwendergemeinschaft für Software-Metrik und Aufwandschätzung

http://www.dasma.org

ISBSG
Log in
SEARCH
Home
Industry Data
Data Portal
Reports & Services
Industry Tools
Academic
About Us
Member Countries
Contact
Purchase ISBSG products
Subscribe and receive exclusive benefits
Sign up for our free newsletter
Submit data and be rewarded
Software Development & Enhancement
Software Maintenance & Support
SEER

The global and independent source of data and analysis for the IT industry

Use industry history data to improve your IT management
Click on your area of interest below

Software Development & Enhancement
Improve IT performance through estimation, benchmarking, project planning & management and IT infrastructure planning.

Software Maintenance & Support
Improve management performance of software portfolio maintenance and support.

"For many years the lack of readily available benchmark data blinded software developers and managers to the real economics of software. Now that ISBSG is making data on thousands of projects available to the software industry, it is becoming possible to make solid business decisions about software development practices and their results in terms of productivity and quality. ISBSG data is a valuable asset for the software industry and for all companies that produce software." Capers Jones

The ISBSG thanks the following supporting organisations

Case Studies

Large Bank: Netherlands
The ISBSG data helped this organization to estimate their project in a realistic way, preventing them from a huge failure. View case study »

News

New Report Performance by Country
3 April 2012
Using the 6,000 projects in the repository we take a look at the characteristics of software

International Software Benchmarking Standard Group (ISBSG)

https://www.isbsg.org

In English

FiSMA – For better Management

Finnish Software Measurement Association FiSMA is a non-profit, independent association focusing on better management through improving the quality and measurability of software & systems engineering and IT service management.

FiSMA's membership is intended for all companies, research units, universities and other institutes interested in software measurement. At the moment, there are about 40 active member organisations and local software process improvement networks (SPINs).

FiSMA is actively participating standardisation activities and perform as a national body of Finland in developing standards under ISO/IEC JTC1 SC7, Software and Systems Engineering subcommittee.

Areas of standardisation which FiSMA follows particularly:

- Software and systems engineering frameworks (ISO/IEC 12207, ISO/IEC 15288, ISO/IEC 15504, CMMI)
- Measurement of software projects (ISO/IEC 14143, ISO/IEC 29881)

Research Forum 1_2015
3.3.2015 klo 12:15-12:15

Scope Manager Forum kokous
10.3.2015 klo 13-16

22nd EuroAsiaSPI (EuroSPI) Conference 2015, call for papers 25.5.2015

Finnish Software Measurement Association (FISMA)

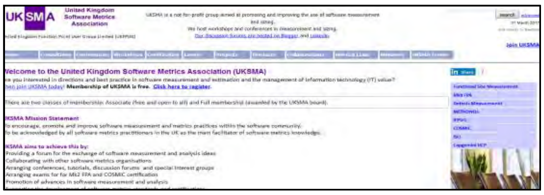
http://www.fisma.fi/in-english/

AEMES was born in 1997 as a Spanish Software Metrics Association (SSMA) and a member of IFPUG and ISBSG, and was joined in 2009 by SFPUG (Spanish FP User Group). Initially oriented to FSM methods such as IFPUG FPA, AEMES progressively covered and included IT Governance, IT financial management of portfolio and projects, risk management and new IT management models. Deep changes are on the way with new elections last May, new board and directions to be confirmed (not only SW) but also metrics.
Members are currently 46 including 6 universities. The current fee for companies are : 360€+VAT/year (plus 510€ for a first time assessment). Linked with similar associations in Latin America. Services to members: annual conference, technical workshops, 1-2 seminars per year and working groups.

Documentation and work products are available only to members by its website. Participation in ISO/IEC JTC1/SC7/WG12 for translating ISO guides and/or standards (e.g. IFPUG FPA, COSMIC).
Quarterly Process and Metrics Review – electronic publication. Training courses on several topics by AEMES partners.
Training courses are provided with fees to members, with some discounts (e.g. for FISMA members).
Presented results from an internal survey on the interest and maturity in adopting certain measurement-related issues for designing the series of courses for a training plan within AEMES.(http://www.aemes.org)

Asociacion Espanola de Metricas de Software

http://www.aemes.org/



UKSMA is a not-for-profit group aimed at promoting and improving the use of software measurement and sizing.
We host workshops and conferences in measurement and sizing.
Our discussion forums are hosted on Blogger and LinkedIn.

Velcome to the United Kingdom Software Metrics Association (UKSMA)
re you interested in directions and best practice in software measurement and estimation and the management of information technology (IT) value?
hen join UKSMA today! **Membership of UKSMA is free. Click here to register.**

There are two classes of membership: Associate (free and open to all) and Full membership (awarded by the UKSMA board).

KSMA Mission Statement
To encourage, promote and improve software measurement and metrics practices within the software community.
To be acknowledged by all software metrics practitioners in the UK as the main facilitator of software metrics knowledge.

KSMA aims to achieve this by:
Providing a forum for the exchange of software measurement and analysis ideas
Collaborating with other software metrics organisations
Arranging conferences, tutorials, discussion forums and special interest groups
Arranging exams for for Mk2 FPA and COSMIC certification
Promotion of advances in software measurement and analysis

United Kongdom Software Metrics Association (UKSMA)

http://www.uksma.co.uk

Gruppo Utenti Function Point Italia
Italian Software Metrics Association

Gruppo Utenti Function Point Italia Italian Software Metrics Association
Il GUFPI-ISMA è l'associazione italiana per la promozione, la diffusione e lo sviluppo delle tecniche quantitative di misurazione del software, inclusi i metodi di misurazione della dimensione funzionale Function Point IFPUG e COSMIC.

Cookie di altre società
GUFPI-ISMA integra, all'interno delle proprie pagine, servizi di terze parti che potrebbero impostare e utilizzare propri cookie e/o tecnologie similari. L'impiego di tali cookie e tecnologie similari da parte di tali aziende è regolato dalle informative sulla privacy di dette società e non dalla presente informativa essendo GUFPI-ISMA totalmente estraneo alla gestione di tali strumenti ed al trattamento dei dati da questi derivanti. Forniamo di seguito un elenco (non esaustivo) di alcune delle società partner che potrebbero utilizzare i cookie mentre navighi sul network di GUFPI-ISMA.
* Facebook (informativa)
* Twitter (informativa)
* LinkedIn (informativa)
* YouTube (informativa)

Gruppo Utenti Function Point Italia - Italian Software Metrics Association (GUFPI - ISMA)

http://www.gufpi-isma.org

Ankündigung ASQT 2017

Die 15. Auflage der Anwenderkonferenz Softwarequalität und Test (ASQT) wird **am 9. und 10. November 2017** an der Technischen Universität Graz ausgerichtet.

ASQT 2016 Vortragsunterlagen

Die 14. Auflage der ASQT fand am 22. und 23. September 2016 an der Universität

Anwenderkonferenz Software-qualität und Test (ASQT)

http://www.asqt.org

MEASUREMENT SERVICES

Software Measurement Laboratory (SML@b)

http://www.smlab.de

INTERNATIONAL FUNCTION POINT USERS GROUP
IFPUG SNAP Case Study Award Program
SNAP
Software Non-Functional Assessment Process
MAKING SOFTWARE SIZE COMPLETE

International Function Point Users Group (IFPUG)

http:www.ifpug.org

Welcome to the Official PSM Web Site!

PSM Users' Group

Additional Information

Practical Software and Systems Measurement (PSM) was developed to meet today's software and system technical and management challenges. It is an information-driven measurement process that addresses the unique technical and business goals of an organization. The guidance in PSM represents the best practices used by measurement professionals within the software and system acquisition and engineering communities. PSM:

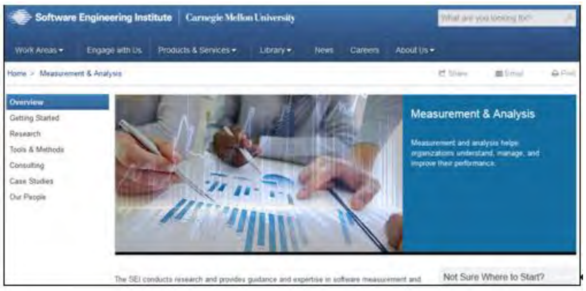
- *Is sponsored by the Department of Defense and the US Army*
- *Provides Project Managers with objective information needed to successfully meet cost, schedule, and technical objectives*

Practical Software & Systems Measurement

www.psmsc.com/:

Computer Measurement Group (CMG)

http://www.cmg.org



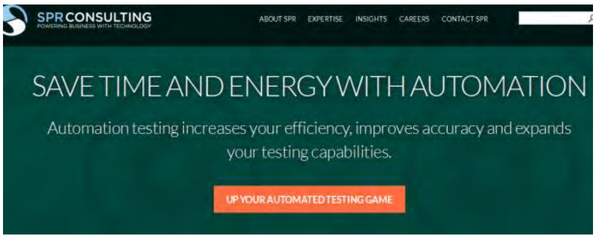
Measurement & Analysis

Measurement and analysis helps organizations understand, manage, and improve their performance.

The SEI conducts research and provides guidance and expertise in software measurement and

Software Engineering Institute (SEI)

www.sei.cmu.edu/measurement/


SPR CONSULTING
POWERING BUSINESS WITH TECHNOLOGY
ABOUT SPR
EXPERTISE
INSIGHTS
CAREERS
CONTACT SPR
UP YOUR AUTOMATED TESTING GAME

SAVE TIME AND ENERGY WITH AUTOMATION

Automation testing increases your efficiency, improves accuracy and expands your testing capabilities.

Software Productivity Research (SPR)

http://www.spr.com/

McCabe SOFTWARE
The Software Path Analysis Company
Call 800-638-6316 or Contact Us Here
Search
YOU ~~WANT~~ NEED THE BEST CODE POSSIBLE
HIGH COMPLEXITY IS ...BAD. LOW COVERAGE IS ...BAD. INSECURE CODE IS ...BAD.
THERE'S ONLY ONE TOOL FOR HELP IN ALL THESE AREAS.
McCabe IQ > Enterprise Edition
30 DAY FREE TRIAL
Our Products
Our Partners
News & Events
About Us
Support
Contact Us
highlights

NEWS: McCabe's University Program Brings Software Quality and Security Solutions into the Classroom

NEWS: Cyber Security: Combat Software Security Risks with McCabe and Bishop Consulting

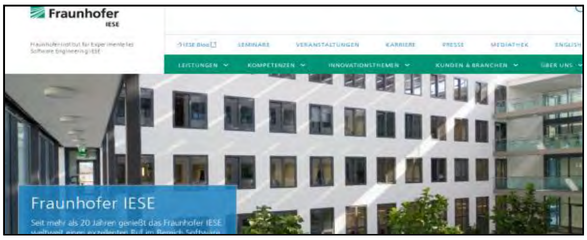
PAPER: Security Risk Identification: 12 Application Architecture Categories to Review

GET STARTED TODAY!
McCABE IQ DEVELOPER'S EDITION
$4,500*
CONTACT A REPRESENTATIVE FOR DETAILS
*Valid for new customers. Restrictions apply.

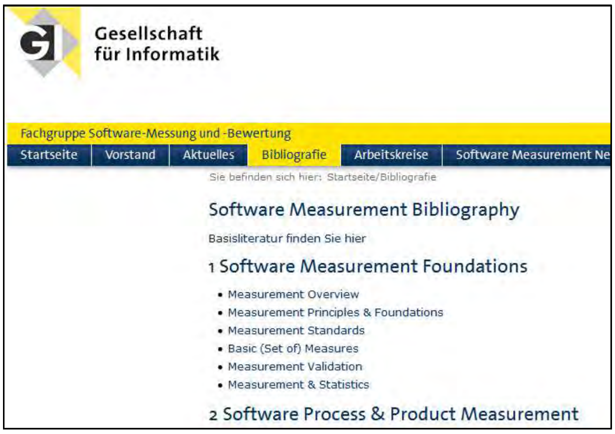
McCabe Software provides software security, quality, testing, release, and configuration management solutions to top commercial software, finance, defense, aerospace, healthcare, and telecommunication providers.

McCabe & Associates

http://www.mccabe.com

 Transforming the World Through Quality Specialist Consultancy Unsere erfahrenen Experten unterstützen Sie bei Ihren Global Delivery Centres Wir liefern Ihnen weltweit die richtigen Lösungen aus unseren Von agilen Methoden profitieren Wie Software-Qualität schneller an Unser Blog Neuigkeiten von SQS auf unserem Blog >>	**SQS Gesellschaft für Software-Qualitätssicherung** ***http://www.sqs.de***
 Navigate with Confidence QSM provides estimation and business analytics to manage your software portfolio investments.	**Quantitative Software Management (QSM)** ***http://www.qsm.com/***
 Fraunhofer IESE Seit mehr als 20 Jahren genießt das Fraunhofer IESE	**Fraunhofer Institute for Experimental Software Engineering (IESE)** **https://www.iese.fraunhofer.de/**
 ENGINEERING LABORATORY The Engineering Laboratory promotes U.S. innovation and industrial competitiveness by advancing measurement science, standards, and technology for engineered systems in ways that enhance economic security and improve quality of life. NIST Awards $20 Million for Research Center to Help Communities Increase Resilience to Disaster The Internet's Next Big Idea: Connecting People, Information, and Things NIST Issues Final Joplin Tornado Report, Begins Effort to Improve Standards and Codes	**National Institute of Standards and Technology (NIST)** ***https://www.nist.gov/el***

SOFTWARE MEASUREMENT INFORMATION



Gesellschaft für Informatik

Fachgruppe Software-Messung und -Bewertung

Sie befinden sich hier: Startseite/Bibliografie

Software Measurement Bibliography

Basisliteratur finden Sie hier

1 Software Measurement Foundations

- Measurement Overview
- Measurement Principles & Foundations
- Measurement Standards
- Basic (Set of) Measures
- Measurement Validation
- Measurement & Statistics

2 Software Process & Product Measurement

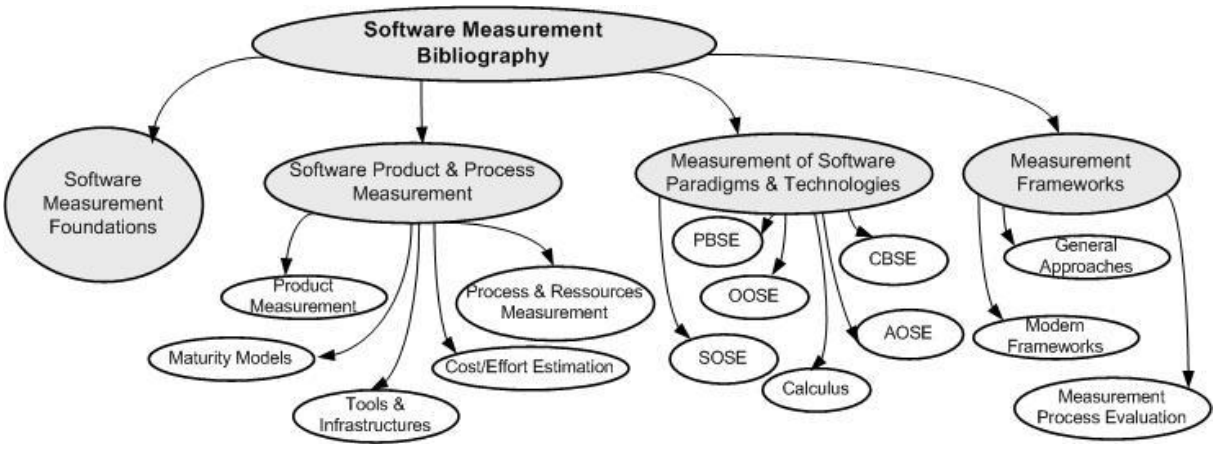
Software Measurement Bibliography

See our overview about software metrics and measurement in the Bibliography at

https://fg-metriken.gi.de/bibliographie/

including any hundreds of books and papers

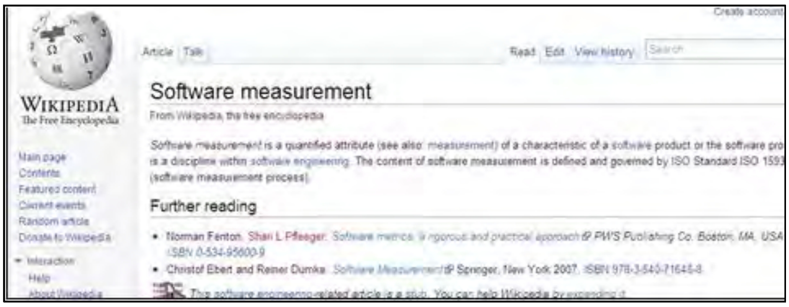
Bibliography Structure:



Software Measurement Bibliography
Software Measurement Foundations
Software Product & Process Measurement
Product Measurement
Process & Ressources Measurement
Maturity Models
Cost/Effort Estimation
Tools & Infrastructures
Measurement of Software Paradigms & Technologies
PBSE
CBSE
OOSE
AOSE
SOSE
Calculus
Measurement Frameworks
General Approaches
Modern Frameworks
Measurement Process Evaluation

Software Measurement & Wikipedia

Help to qualify the software measurement knowledge and intentions in the world wide web:



Software measurement

From Wikipedia, the free encyclopedia

Software measurement is a quantified attribute (see also: measurement) of a characteristic of a software product or the software pro is a discipline within software engineering. The content of software measurement is defined and governed by ISO Standard ISO 1593 (software measurement process).

Further reading

- Norman Fenton, Shari L Pfleeger: *Software metrics: a rigorous and practical approach* PWS Publishing Co. Boston, MA, USA ISBN 0-534-95600-9
- Christof Ebert and Reiner Dumke: *Software Measurement* Springer, New York 2007, ISBN 978-3-540-71648-8

This software engineering-related article is a stub. You can help Wikipedia by expanding it.

Software metric

From Wikipedia, the free encyclopedia

A **software metric** is a measure of some property of a piece of software or its specifications. Since quantitative measurements are essential in all sciences, there is a continuous effort by computer science practitioners and theoreticians to bring similar approaches to software development. The goal is obtaining objective, reproducible and quantifiable measurements, which ma have numerous valuable applications in schedule and budget planning, cost estimation, quality assurance testing, software debugging, software performance optimization, and optimal perso task assignments.

Contents [hide]

1 Common software measurements
2 Limitations
3 Acceptance and public opinion
4 See also
5 References
6 External links

Common software measurements [edit]

Common software measurements include:

- Balanced scorecard
- Bugs per line of code
- Code coverage
- Cohesion

Coming soon: a new SWEBOK Guide

Volunteers are in the process of refreshing the Guide to the Software Engineering Body of Knowledge (SWEBOK)—adding new knowledge areas (KAs) and revising others. For the latest materials available for public review as well as guidelines for reviewing, visit the SWEBOK V3 Public Review site.

Twelve KAs have completed public review:

- Software Engineering Economics
- Software Requirements

SWEBOK news

3 New KAs in SWEBOK V3 Open for Public Review
The Software Engineering Economics, Software Requirements, and Software Testing Knowledge areas for SWEBOK V3 are now available for public review.

> Read more

Software Engineering Body of Knowledge (SWEBOK)

http://***www.swebok.org***

PMBOK® Guide and Standards

Achieve Excellence
PMI provides resources and global standards as a foundation for the profession.

Project Management Body of Knowledge (PMBOK)

http://www.pmbook.org

Software Measurement News

VOLUME 27 2022 NUMBER 2

CONTENTS

ISSN 1867-9196