How can Requirements Engineers help Project Estimators?

Charles Symons
Co-Founder & Past President, COSMIC

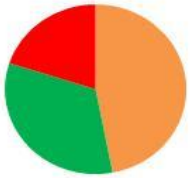
BCS Requirements Engineering Specialist Group, 11th March, 2015

Agenda

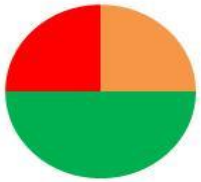
- The economic importance of the RE/Estimator relationship
- Basics of project estimating and of sizing requirements
- Non-functional requirements
- Mapping of RE methods to Functional Size Measurement methods
- Some observations & conclusions

Project delivery to time & budget is notoriously poor

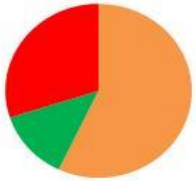
Typical Standish findings [1]



European Union study '98 - '05 [2]



UK Public Sector report 2007 [3]



ISBSG 2001 - '08 [4]

Failed

>10% over budget

Successful

>20% over budget

'Successful'

(Trend is improving)

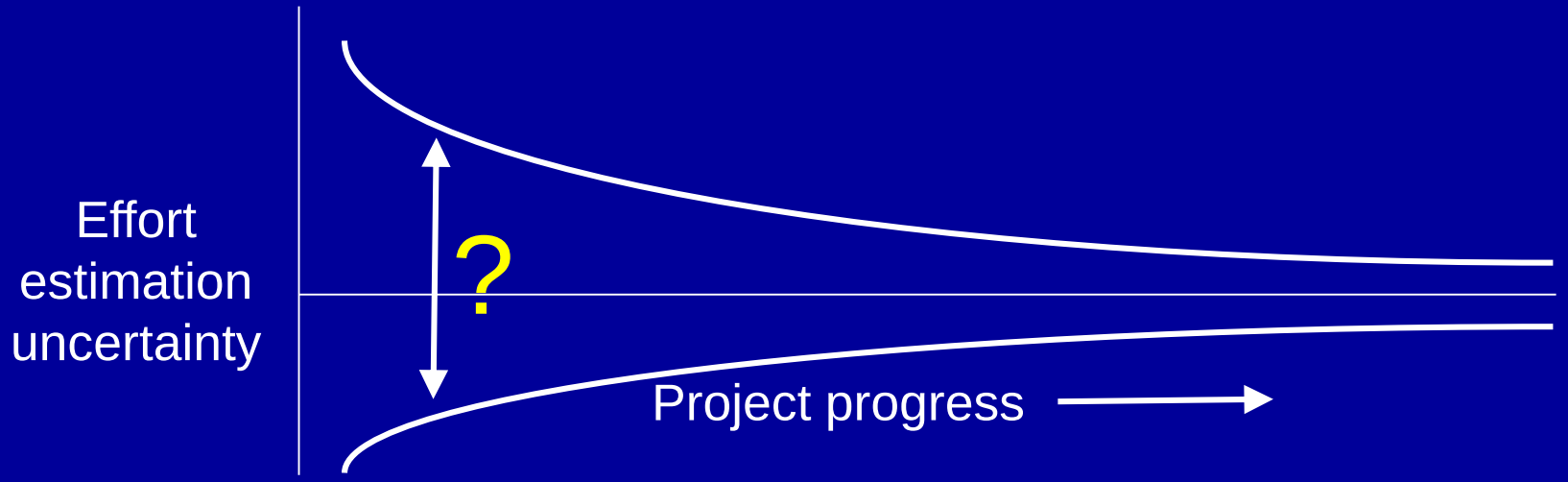
Why do we get these over-runs and failures?

Demand side:

Lack of senior management commitment & user involvement → Stakeholder conflicts; unclear objectives → **Vague and shifting requirements** → Developer estimates cost & time, and adds contingency

Supply side:
Poor project management (e.g. of progress and changes),
staff inexperience, especially with new technology, staff turnover, etc

Estimating total development effort early in a project life is intrinsically difficult …



…. but is usually needed for the investment decision, and is usually politically difficult to change

… and project effort estimation is an immature subject

- Most estimating is by analogy or by expert judgement [5]
  - (consequently mostly optimistic)
- No standard estimating methods

Agenda

- The economic importance of the RE/Estimator relationship
- Basics of project estimating and of sizing requirements
- Non-functional requirements
- Mapping of RE methods to Functional Size Measurement methods
- Some observations & conclusions

Serious project effort estimation needs a measure of software size and data on past performance

$$\text{Development Productivity} = \frac{\text{Software Size}}{\text{Effort}}$$

How big it it?

$$\text{Estimated development effort for 'New-Project'} = \left(\frac{\text{Est. Soft. Size for 'New-Project'}}{\text{Past Productivity}}\right) \times \left(\text{Adjustments for 'New-Project'}\right)$$

Software size is the biggest driver of project effort

Sizing method options:

SLOC:

- ❖ Can't estimate until software designed
- ❖ Technology-dependent, no standards
- ✓ Accounts for all requirements

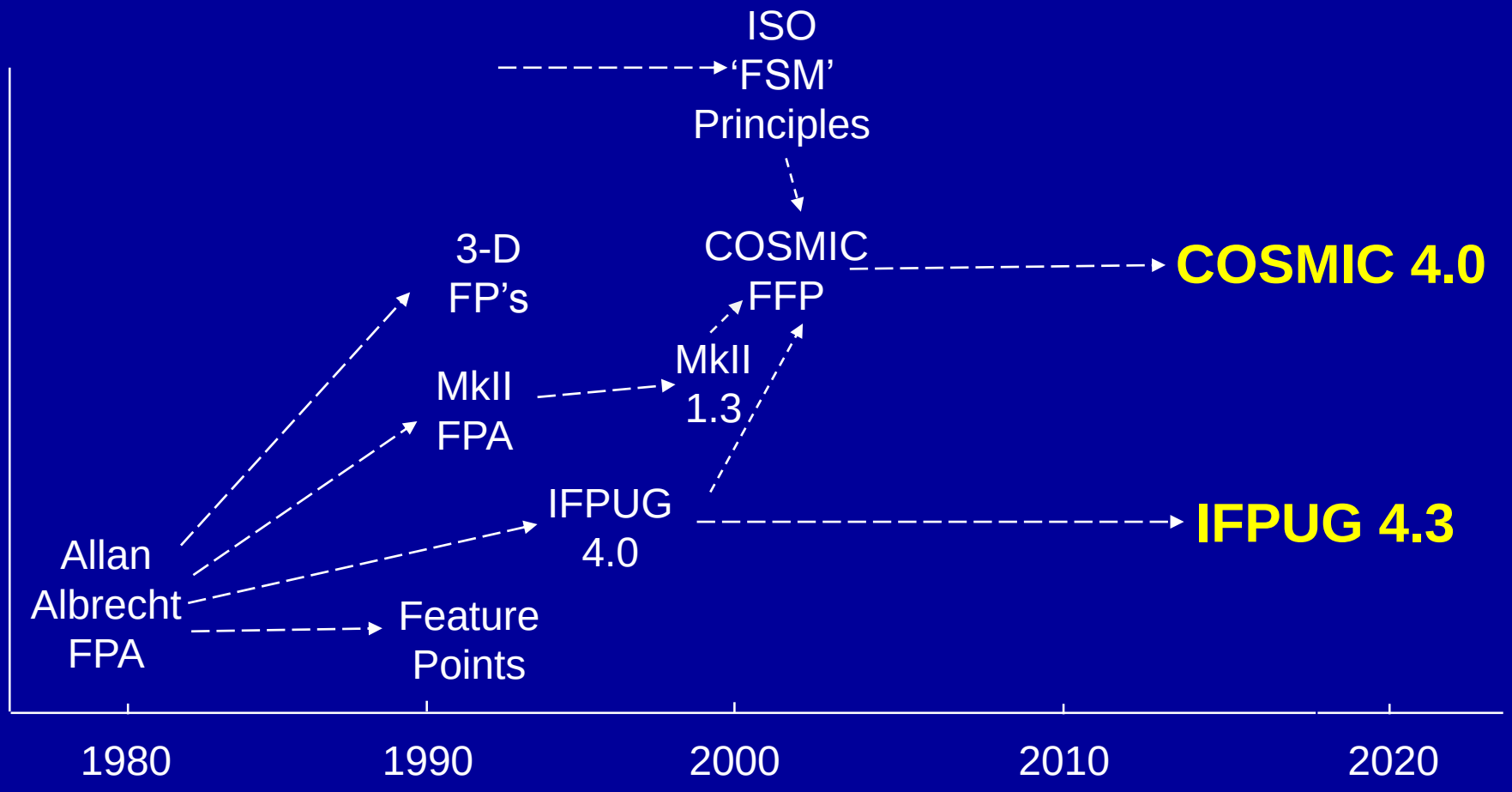
Functional size:

- ✓ International standard methods:
- ❖ What about Non-Functional Requts?

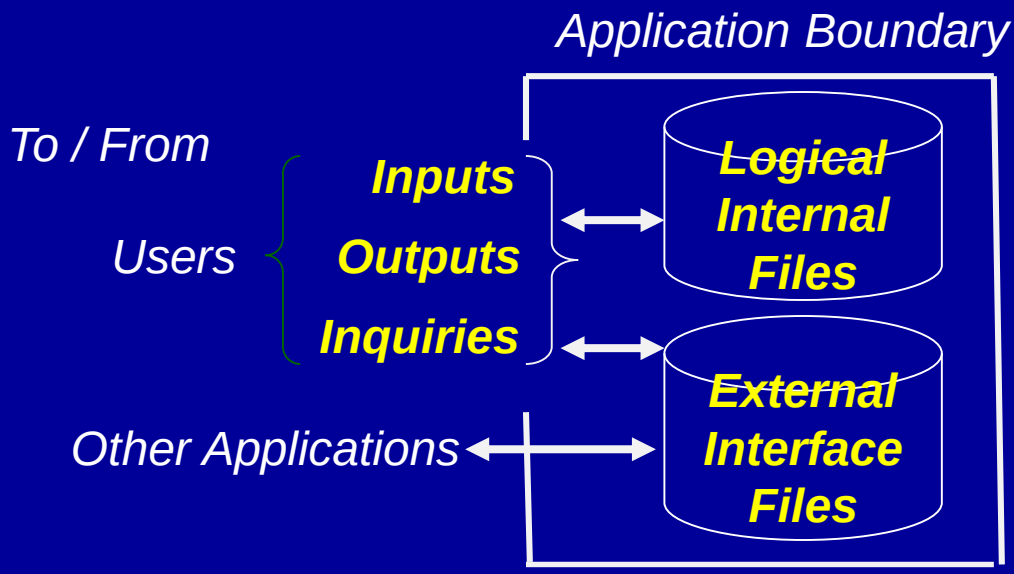
Other sizing methods
e.g. UCP, OOP, SP, etc:

- ❖ No reliable standards
- ❖ (So no publicly-available benchmarks)

There are only two significant ISO standard methods of measuring a Functional Size



Albrecht/IFPUG Function Points – based on a 1970's model of software [6]



14 x 'General Application Characteristics'

Obsoelete

Size (Function Points) = "Unadjusted Function Points" X "Value Adjustment Factor"

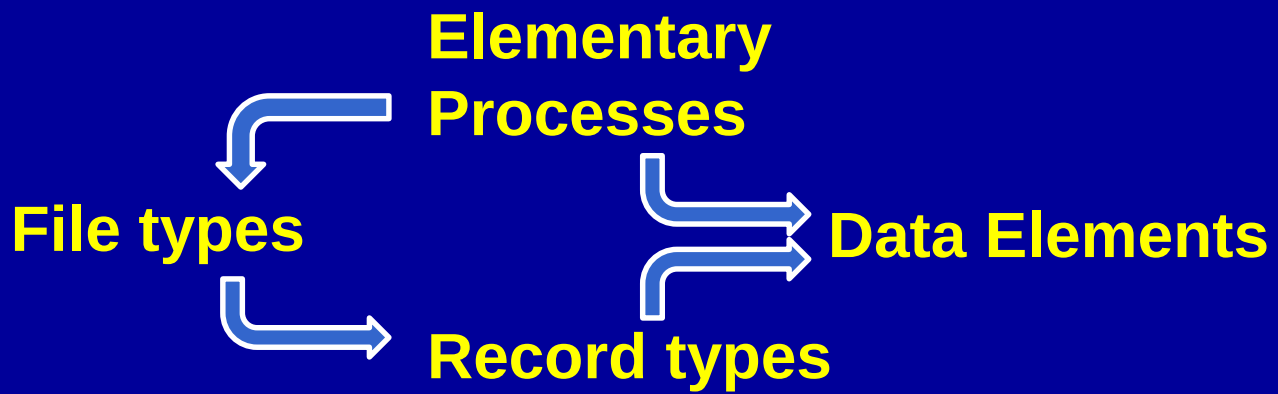
Information Processing Size

Adjustment for Technical & Quality (NF) Requirements

What an Estimator needs to know from the requirements to measure a FP size:

Functional Component	Size (Function Points) Simple	Av.	Complex	Complexity Depends on
External Input	3	4	6	# File types referenced # Data Element types
External Output	4	5	7	
External Inquiry	3	4	6	
Logical Internal File	7	10	15	# Record types # Data Element types
Ext. Interface File	5	7	10	

IFPUG Software model: Summary



Measurement Rules

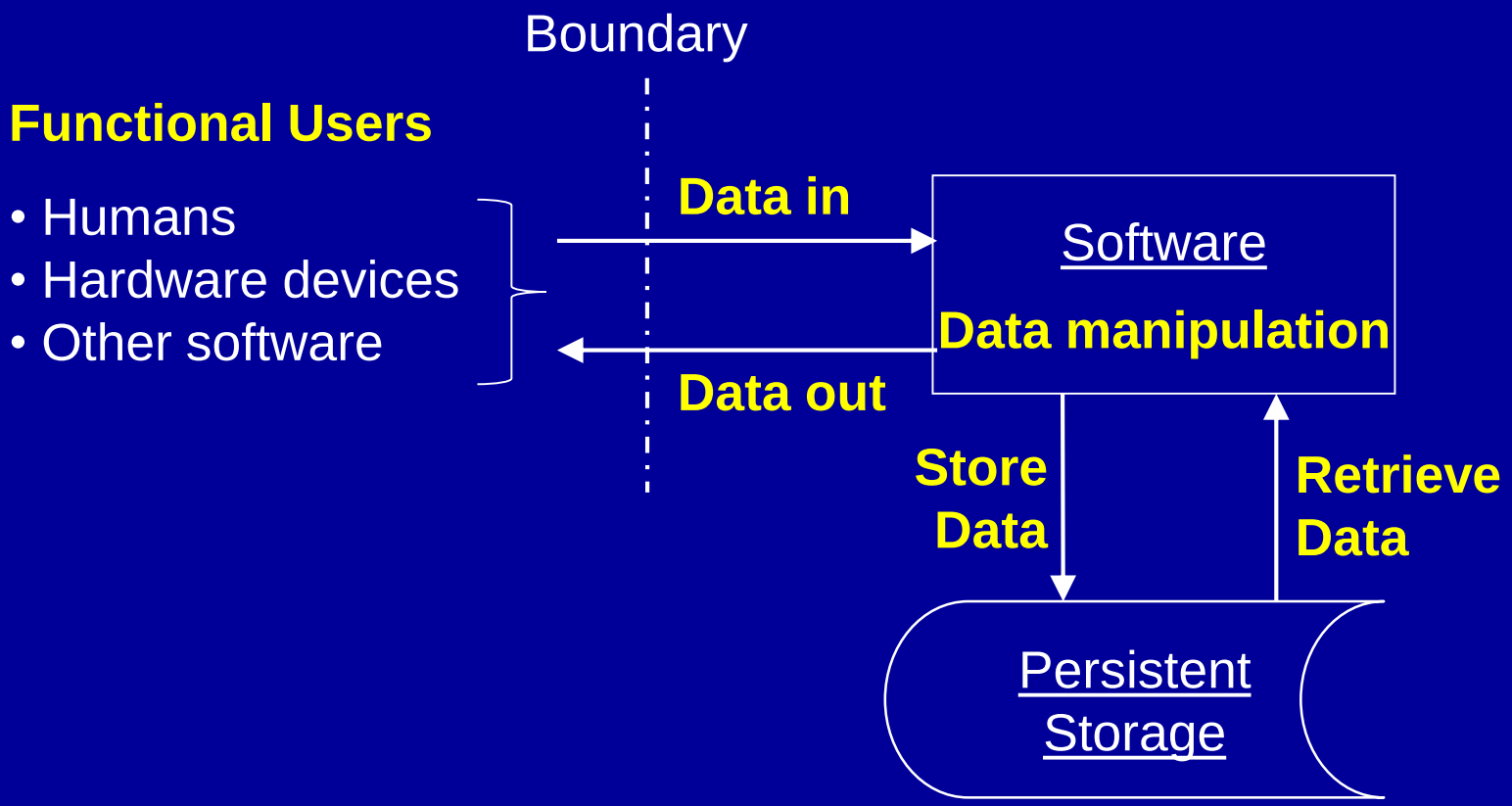
Size (Unadjusted FP's) = Σ Size (Elementary Processes)
+
Σ Size (File types)

REMARKS:

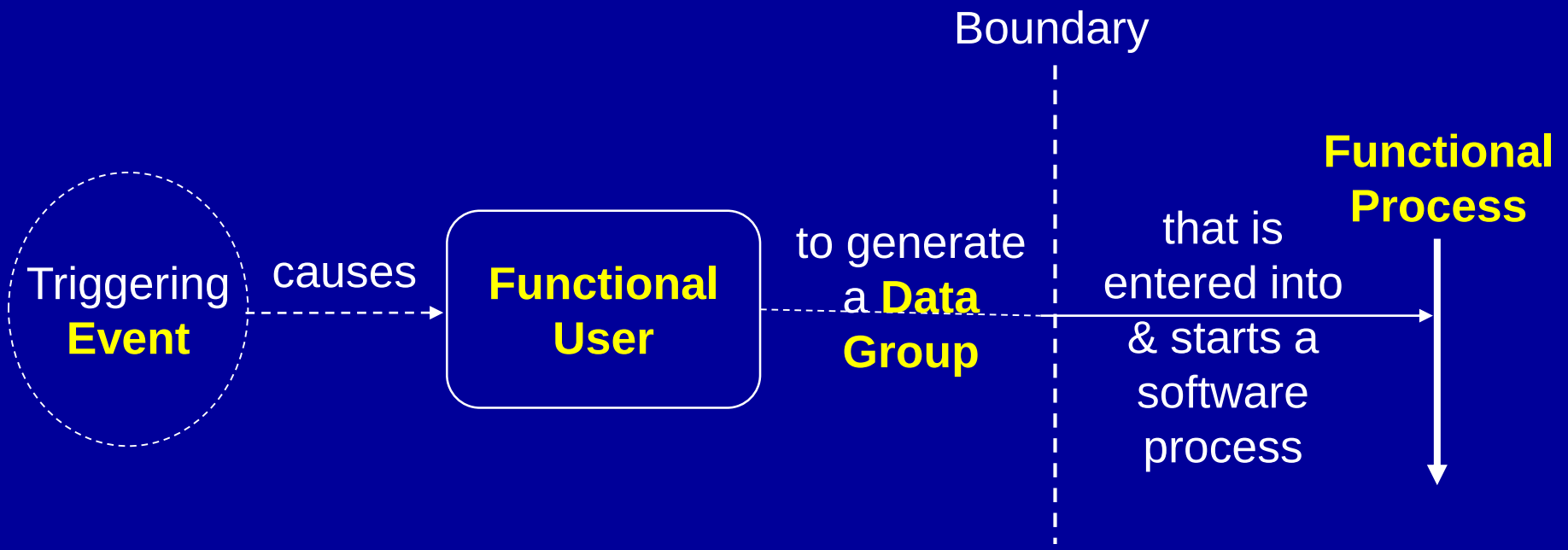
- The IFPUG model was designed to measure business applications
- A precise FP size measurement needs detailed requirements

COSMIC method – based on fundamental software engineering principles [7]

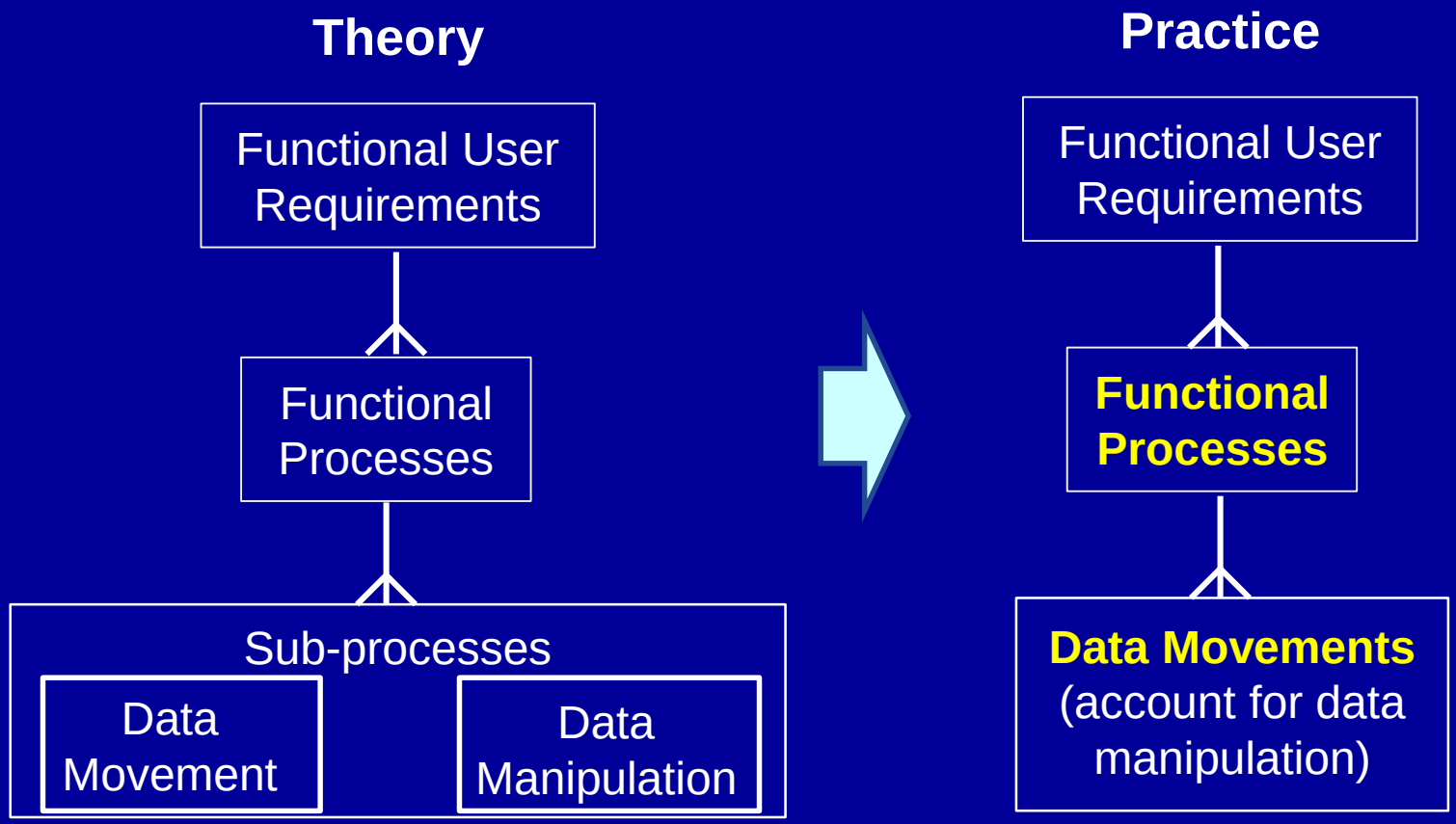
1. Generic Software Model:



2. An event causes a ‘Functional User’ to trigger a ‘Functional Process’ in the software

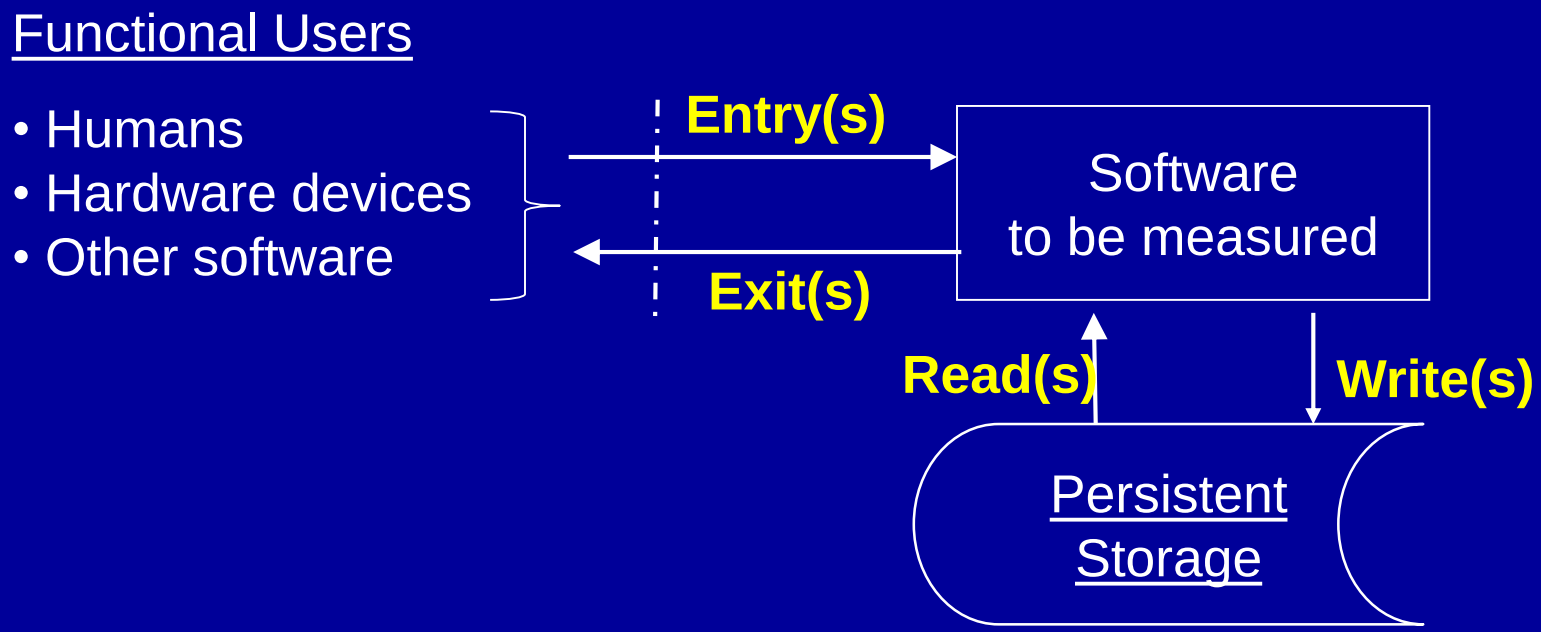


3. Functional Processes can be decomposed into ‘Sub-processes’

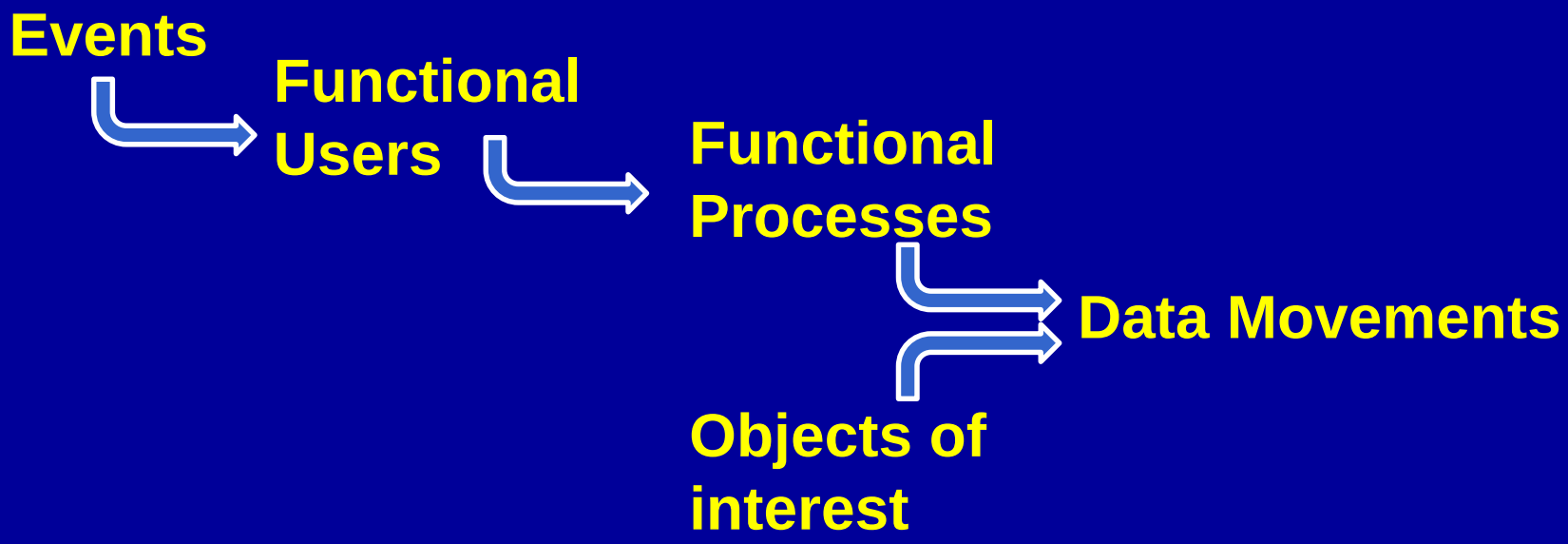


4. A Data Movement moves a ‘Data Group’ that describes one ‘Object of Interest’

- An ‘**OOI**’: *a thing in the world of the user about which the software is required to move or store data*
- There are four Data Movement types:



What an Estimator needs to know from the requirements to measure a CFP size:



COSMIC Measurement Rules

Size (data movement) = 1 x COSMIC FP (CFP)

Size (functional process $_i$) = Σ # Entries$_i$ + Σ # Exits$_i$ + Σ # Reads$_i$ + Σ # Writes$_i$

Size (software) = Σ size (functional process $_i$)

REMARKS:

- The COSMIC model was designed to measure business application, real-time and infrastructure software, at any level of decomposition
- A precise FP size measurement needs detailed requirements

Both methods have variants to measure an approximate size before requirements are known in full detail

Simple Examples

IFPUG: FP = 35 x # ILF's + 15 x # EIF's

COSMIC: CFP = (# Func. Processes) x (Av. CFP size)

Complex Avionics Example [8]

Quartiles of total size ->	Small	Medium	Large	V. Large
Av. CFP size	5.5	11	18	39
% FP's	49%	26%	16%	7%

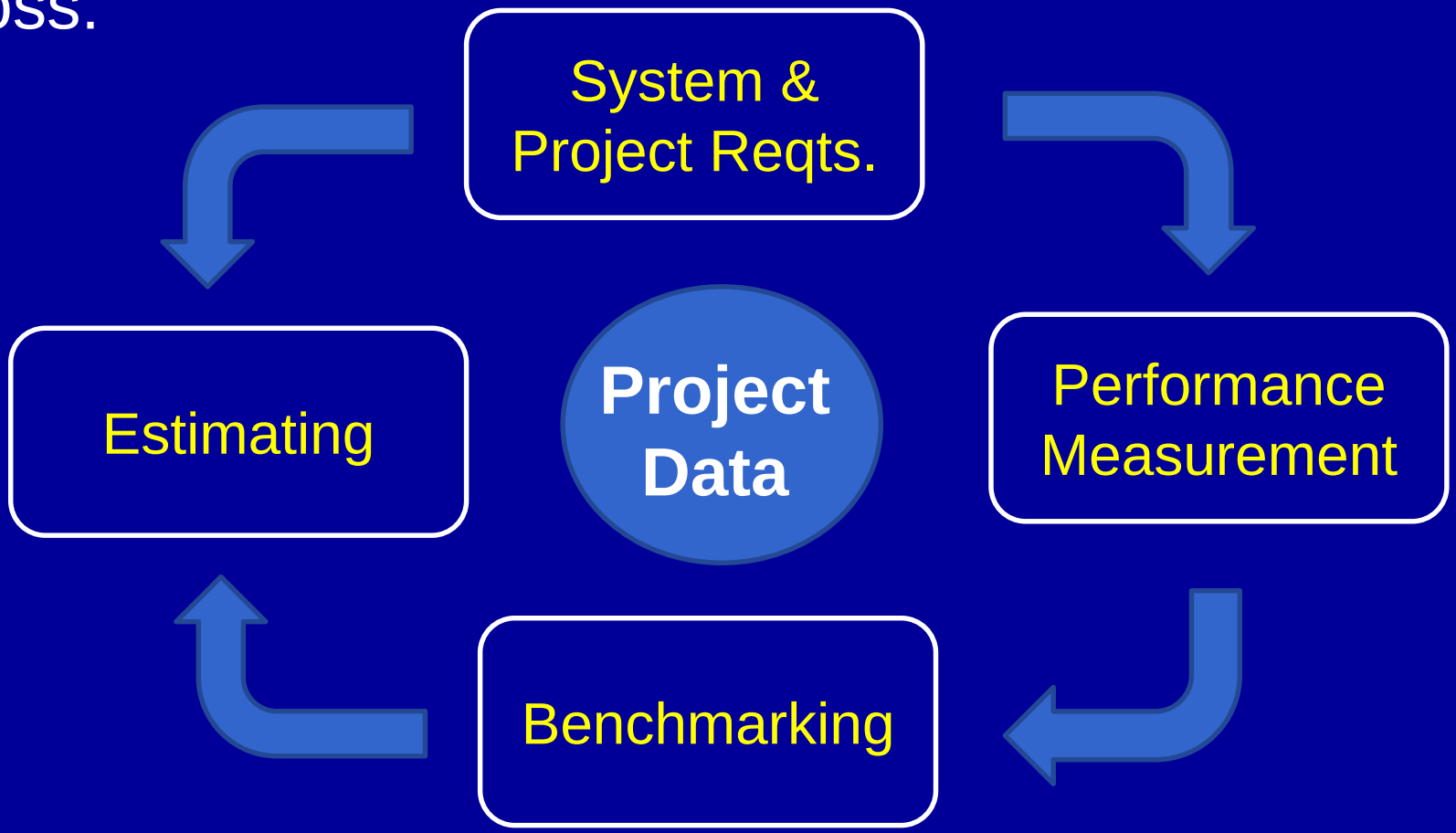
All approximate sizing methods should be calibrated locally

Agenda

- The economic importance of the RE/Estimator relationship
- Basics of project estimating and of sizing requirements
- Non-functional requirements
- Mapping of RE methods to Functional Size Measurement methods
- Some observations & conclusions

There is no good definition, nor any consistent view of what are NFR [9]

Ideally we need consistency across:



Accounting for Non-Functional Requirements: IFPUG proposes a 'SNAP' size measure [10]

NFR Definition: '*Requirements that are not included in the standard IFPUG method, yet need project effort*'

Total estimated project effort =

Effort on FUR = IFPUG FP size / (Functional productivity)

+

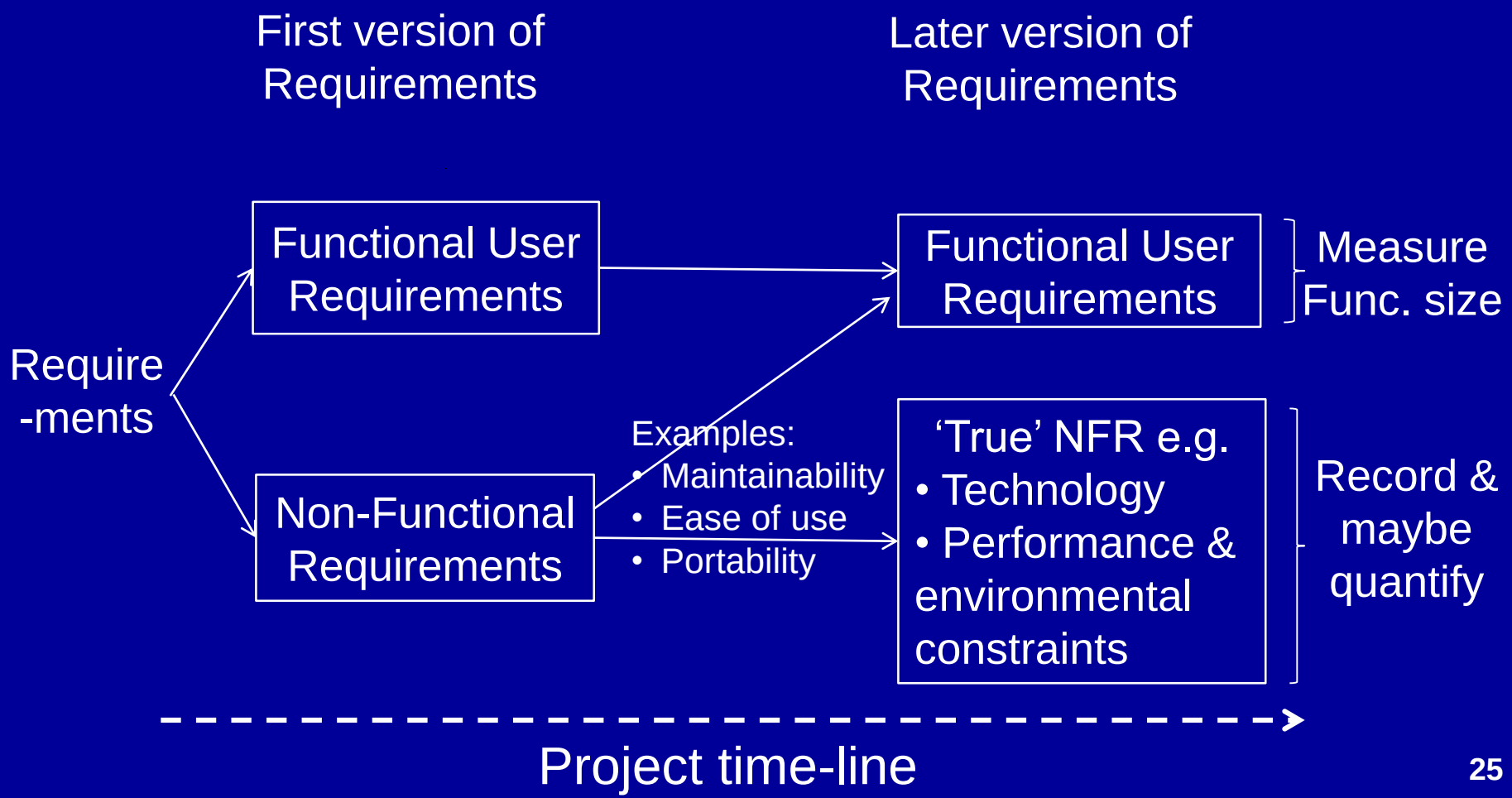
Effort on NFR = SNAP Points size / (NF productivity)

COSMIC does not support any composite size measure for NFR's [11]

NFR Definition: '*Any requirement for:*

- *the software part of a hardware/software system or software product*
- *including how it should be developed and maintained, and how it should perform in operation,*
- *except a functional user requirement for software.'*

As a project progresses, many NFR evolve, wholly or partly into Functional Reqts.



COSMIC/IFPUG are developing a Glossary of standard NF and Project Requirement terms to assist estimating, benchmarking, etc

Main Class	Definition	# Terms
Quasi NFR (System and Software Product Quality Requts.)	Requirements for quality or design that may evolve, wholly or partly, into functional requirements (Grouped using the ISO25020 'SQuaRE' Quality Model)	43
True NFR		
• Environmental Constraints	Characteristics of the environment in which the software must be developed and which it must support.	6
• Technial Constraints	Technology on which the system must run, programming language, etc	11
Project Requts. & Constraints	Characteristics of the project processes, resources, risk, etc	21

The resulting COSMIC approach for project effort estimating….

$$\text{Estimated development effort for 'New-Project'} = \left(\frac{\text{Estimated FS* for 'New-Project'}}{\text{Past Productivity}} \right) \times \left(\begin{array}{c} \text{Adjustments for different} \\ \text{True NFR and project} \\ \text{constraints for 'New-Project'} \end{array} \right)$$

* Functional Size (including the size that originated from NF requirements)

Agenda

- The economic importance of the RE/Estimator relationship
- Basics of project estimating and of sizing requirements
- Non-functional requirements
- Mapping of RE methods to Functional Size Measurement methods
- Some observations & conclusions

Data Models: essential if measuring business applications

- An Entity-type in a E/RA model
- The subject of a Relation in 3NF
- A UML Class

is equivalent to:

- a COSMIC Object of Interest
- an IFPUG Record Type (most likely!)

Note: in the COSMIC method:

Input and Output data must be analysed to identify the OOI's and hence the Entries and Exits, respectively.

UML models: should be easy to measure a functional size, but in practice Use Cases vary enormously.

Company A – Project Type: I

UC No	# of Trans.	FP (Trans. Size)	CFP	CFP / FP
UC1	1	6	27	4.5
UC2	1	7	25	3.6
UC3	1	6	29	4.8
UC4	3	16	46	2.9
UC5	1	6	30	5.0
UC6	1	6	28	4.7
UC7	9	44	112	2.5
UC8	9	59	122	2.1
UC9	2	12	52	4.3
UC10	2	9	25	2.8
UC11	1	6	30	5.0
UC12	15	88	267	3.0
UC13	10	51	113	2.2
UC14	5	17	24	1.4
UC15	1	6	10	1.7

Company A - Project Type: II

UC No	# Trans.	FP (Trans.Size)	CFP	CFP/FP
UC1	1	7	22	3.1
UC2	1	7	13	1.9
UC3	1	7	15	2.1
UC4	1	7	25	3.6
UC5	1	7	17	2.4
UC6	1	7	14	2.0
UC10	1	7	13	1.9
UC11	1	7	18	2.6
UC12	1	7	14	2.0
UC13	1	7	20	2.9
UC14	1	6	17	2.8
UC15	1	7	10	1.4
UC16	1	7	17	2.4
UC17	1	7	15	2.1
UC25	4	24	32	1.3
UC26	4	13	16	1.2
UC27	1	6	8	1.3
UC28	4	12	17	1.4

Different project types may have different:

- **# of Transactions / UC**
- **Average size / UC**

UML: Functional size measurement of 'OTOPOP' Use Cases is easy

- A 'one time, one place, one person' Use Case should be equivalent to:
  - An IFPUG Elementary Process
  - A COSMIC Functional Process

- Measurement of the CFP size of Message Sequence Diagrams for OTOPOP Use Cases is being automated [13]

Finite state machines & EARS syntax map well to COSMIC concepts

- WHEN <trigger>
- IF <trigger>

} Triggering event (may cause one or more state transitions)

- <system response> ≡ functional process
- WHILE <in a specific state> may need a state inspection via:
  - an Entry
  - an Entry/Exit pair
  - a Read

Agile Methods: COSMIC sizing is applicable at any level of aggregation [14]

- **COSMIC** sizing may be applied at user story, iteration, release, whole system levels.
  (In practice, track progress by measuring completed functional processes.)

- Difficult to apply **IFPUG** to Agile deliverables due to aggregation problems

Renault [15] uses CFP sizing to control the development and enhancement of Electronic Control Units (ECU's)

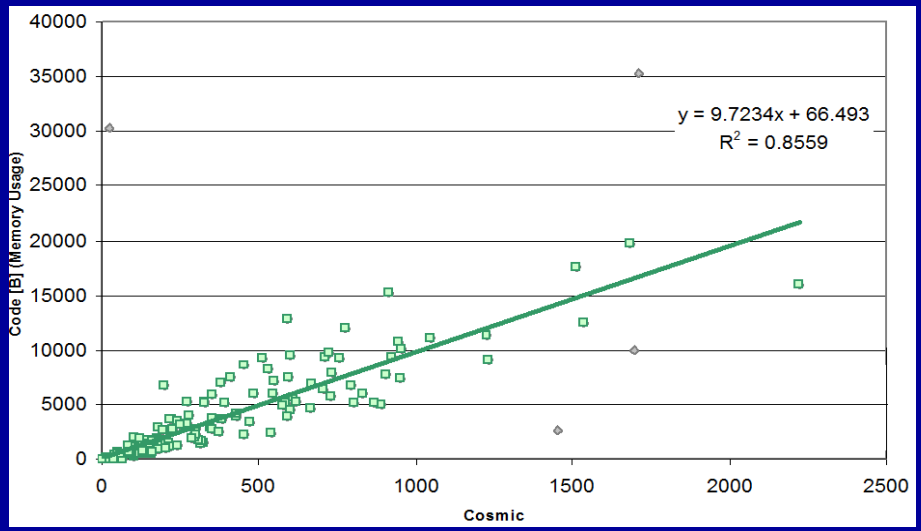
- tracks progress of ECU specification teams…
- who create designs in Matlab Simulink…
- which are automatically measured in CFP

Motivation for automation:- speed and accuracy of measurement

… achieving remarkable cost estimation accuracy from the designs

Cost vs size (CFP)

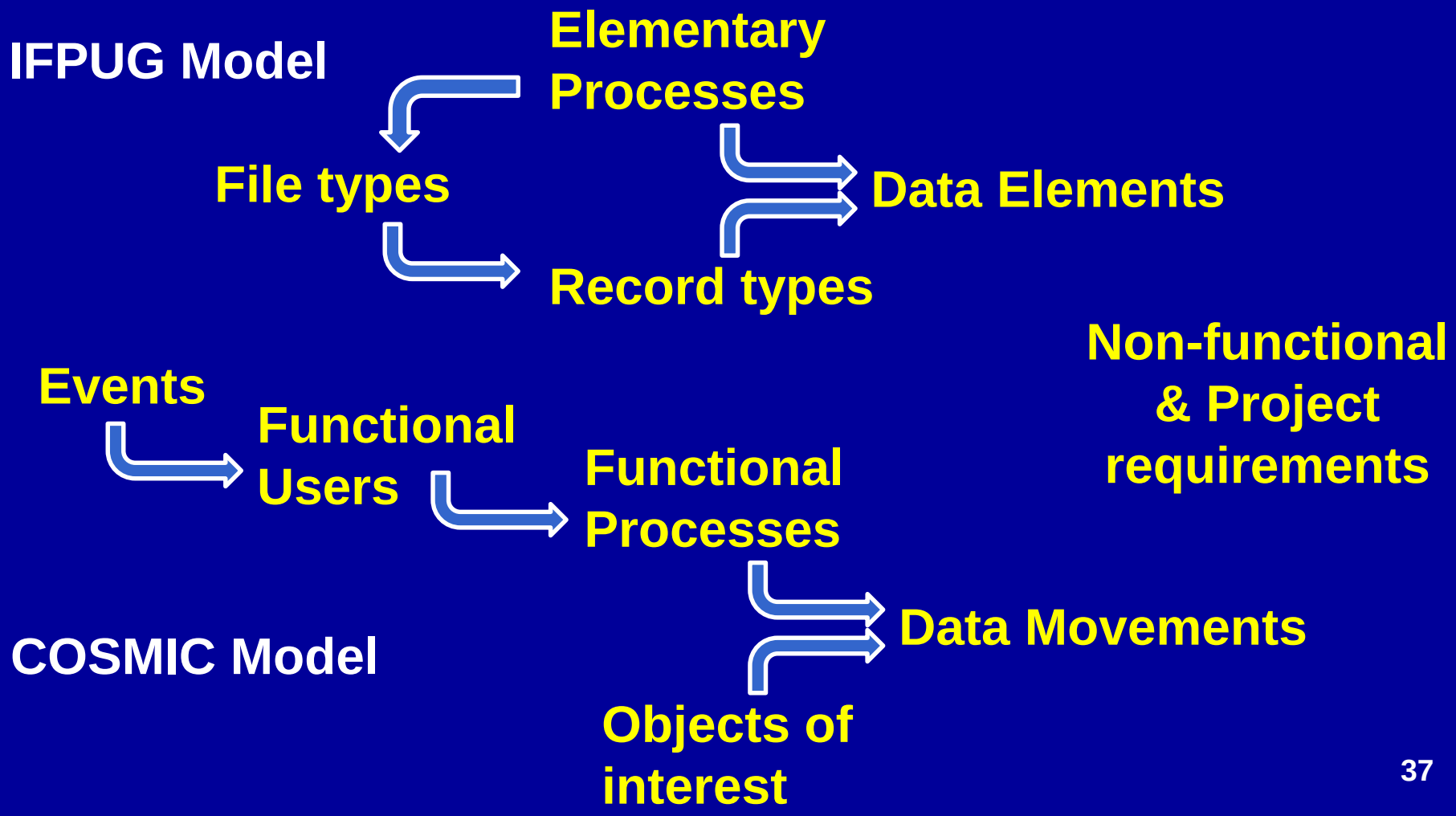
Memory size vs software size (CFP)



Agenda

- The economic importance of the RE/Estimator relationship
- Basics of project estimating and of sizing requirements
- Non-functional requirements
- Mapping of RE methods to Functional Size Measurement methods
- Some observations & conclusions

Project estimation only needs data normally gathered during requirements elicitation



There are many incidental benefits from measuring requirements

- If requirements cannot be measured, the software cannot be built

  (Think of measurement as a QC process)

- Measurements enable:
  - Control of scope creep
  - Understanding and improving project performance
  - More accurate project estimation
  - Contracting on price/unit size, etc

Thank you for your attention

cr.symons@btinternet.com

All COSMIC method documentation is available for free download from
www.cosmic-sizing.org

References

1. Standish CHAOS Report, 2009, www.standishgroup.com/newsroom/chaos_2009.php
2. McManus, J. and Wood-Harper, T., "A Study in Project Failure", www.bcs.org, June 2008
3. Whitfield, D., 'Cost over-runs, delays and terminations: 105 outsourced public sector ICT projects', European Services Strategy Unit, Research Report No. 3, December 2007
4. "The SIP (Software Industry Performance) Report" and ISBSG benchmark data, www.isbsg.org
5. 'A review of studies on expert estimation of software project effort', Magne Jorgensen, Journal of Systems and Software, 70, 2004.
6. Counting Practice Manual, IFPUG 4.3, the international Function Points User Group, www.ifpug.org
7. Measurement Manual, COSMIC 4.0, the Common Software Measurement International Consortium, www.cosmic-sizing.org
8. 'Advanced & Related Topics' (or 'Guideline for approximate COSMIC functional size measurement' – in preparation), www.cosmic-sizing.org
9. 'Accounting for non-functional requirements in productivity measurement, benchmarking & estimating', Symons, C.R., UKSMA Conference 2011, www.uksma.co.uk
10. 'Software non-functional assessment process (SNAP'), www.ifpug.org
11. 'Guideline on how to account for non-functional & project requirements in software project performance measurement, benchmarking & estimating', (in preparation) www.cosmic-sizing.org
12. 'From requirements to project effort estimates – work in progress (still?), Symons, C.R., Gencel, C., REFSQ Conference, Essen, Germany, April 2013
13. 'Automatic COSMIC sizing of requirements held in UML', COSMIC Masterclass Part 3, International Workshop on Software Metrics 2014, Rotterdam, www.ieeexplore.org
14. 'Guideline for use of COSMIC FSM to manage Agile projects', September 2011, www.cosmic-sizing.org
15. 'Manage the automotive embedded software development cost & productivity with the automation of a Functional Size Measurement Method (COSMIC)" Alexandre Oriou et al, IWSM 2014, Rotterdam, www.ieeexplore.org