



“Software industry performance: improving its measurement

Charles Symons

President

The Common Software Measurement International Consortium

SPIN London meeting, 20 January 2011

© Charles Symons, 2011



My messages

- Software industry performance is very poor in some ways, partly due to poor software measurement and estimating practices
- There are solutions and the value of possible improvements is enormous!
- The COSMIC sizing method is part of these solutions. We have made excellent progress over the last 12 years: there is plenty more to do



Software Industry performance is very uneven

- | | | |
|-----------------------------|----------------------|-------------------------------|
| • Delivery to time & budget | Notoriously bad | Estimating difficult |
| • Delivered quality | Often amazingly good | Measurement easy |
| • Productivity | Only slow progress | } Measurement quite difficult |
| • Speed of delivery | No-one knows | |

Poor measurement & estimating methods and processes contribute to poor performance



Some estimates of the annual waste in the software industry

- US software industry: \$100 Billion*
- European software industry €100 Billion**

‘Waste in the software industry’ = write-offs due to project failures, and cost and time over-runs

- carried by software **customers**

* Standish CHAOS Report, 2006, www.standishgroup.com

** McManus, J. and Wood-Harper, T., “A Study in Project Failure”, www.bcs.org, June 2008



How does this waste arise?

Poor project
management

Poor
accountability

etc., etc

Shifting, unclear
requirements

and/or poor **'estimating practices'** ?

- Estimating the size of the software to be built
- The method of converting size to effort and time
- The way in which estimating is integrated with the project management process



The basis of 'top-down' estimating

Productivity = Software size / Project effort

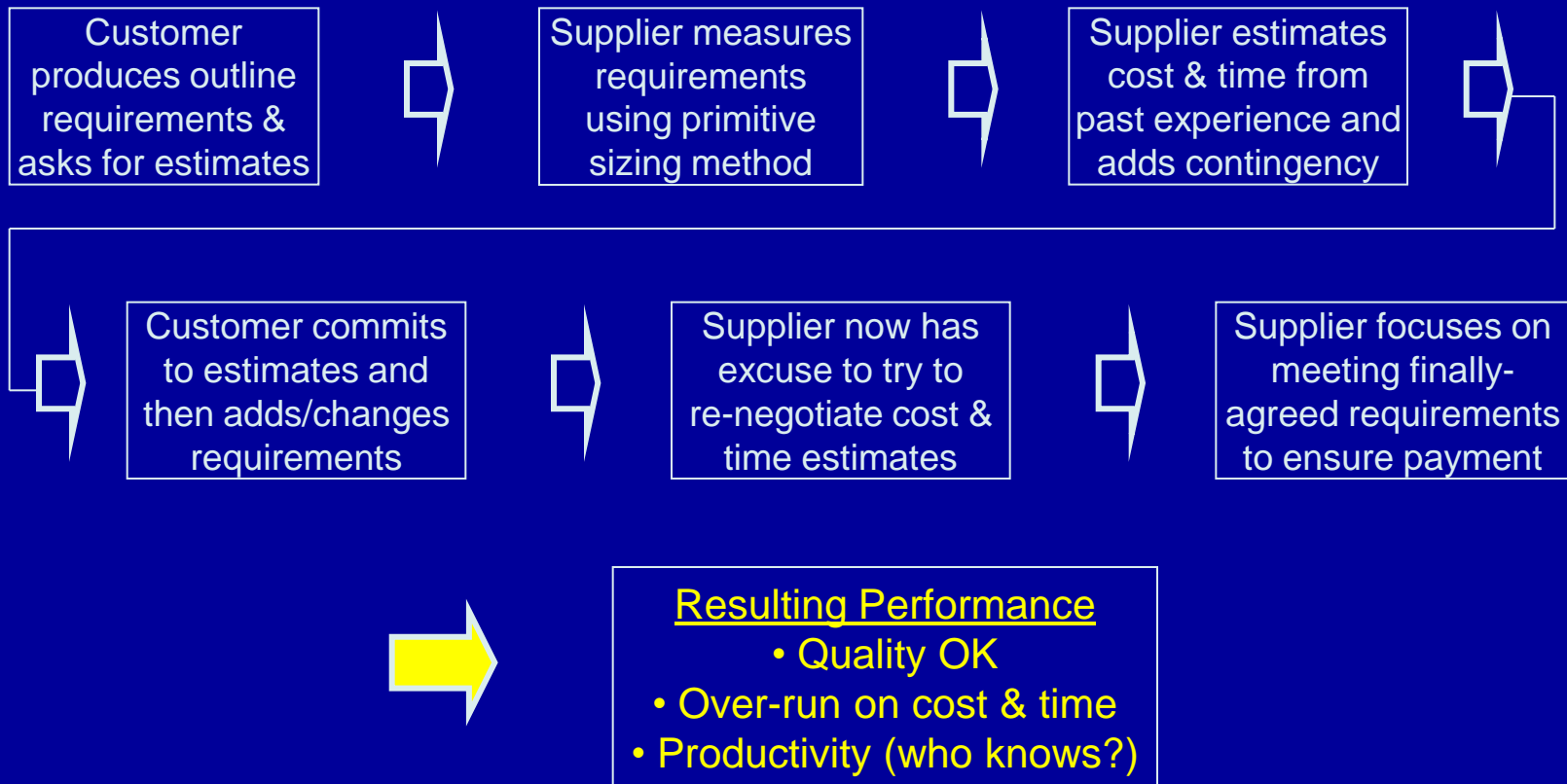
Estimated new project effort = $\frac{\text{Estimated software size}}{\text{Assumed project productivity}}$

Estimated new project effort = $\left\{ \frac{\text{Estimated software size}}{\text{Assumed project productivity}} \right\} \times \left\{ \text{Adjustments for project-specific factors} \right\}$

Conclusion: Reliable software sizing, especially Functional Size Measurement (FSM), is critical for productivity measurement and for estimating methods



Evidence suggests a causal chain linking measurement and estimating to actual performance





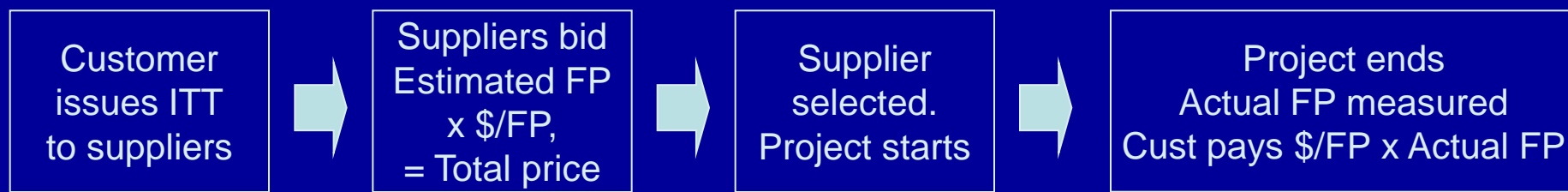
Recommendations for the three elements of 'estimating practices'

- Estimating the size of the software to be built (COSMIC)
- Estimating effort and time starting with size (?)
- The way in which estimating is integrated with the project management process (Southern Scope)



The 'Southern Scope' process uses FSM for successful project control

Process overview:



Results: *

Project management approach	\$/FP	Av. cost over-run
Traditional	1500	84%
Southern Scope	500	<10%

Scope management role (metrics expert) is critical – and highly visible!

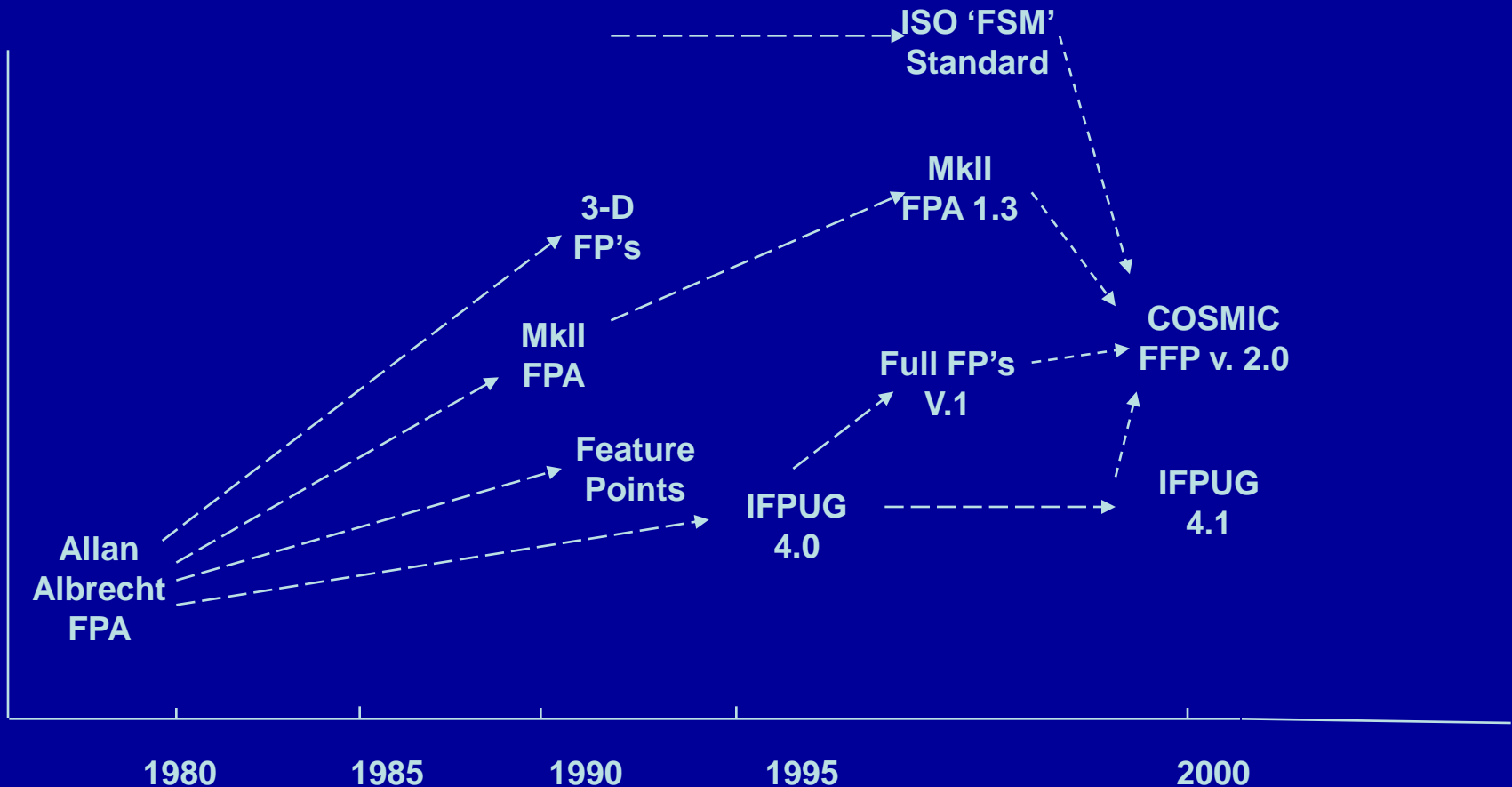
* 'Software development projects in Government; performance, practices and predictions', ISBSG, 2005



**All advances in science &
technology have required
improvements in measurement**



Functional Size Measurement (FSM) methods evolved for >20 years 'BC'





Over the last 12 years the COSMIC method has hardly changed. The support has grown enormously

Software
Context
Model



Generic
Software
Model



Support
Infra-
structure



2000

2002

2004

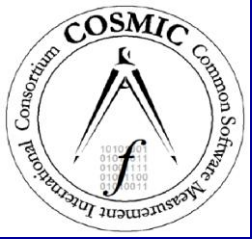
2006

2008

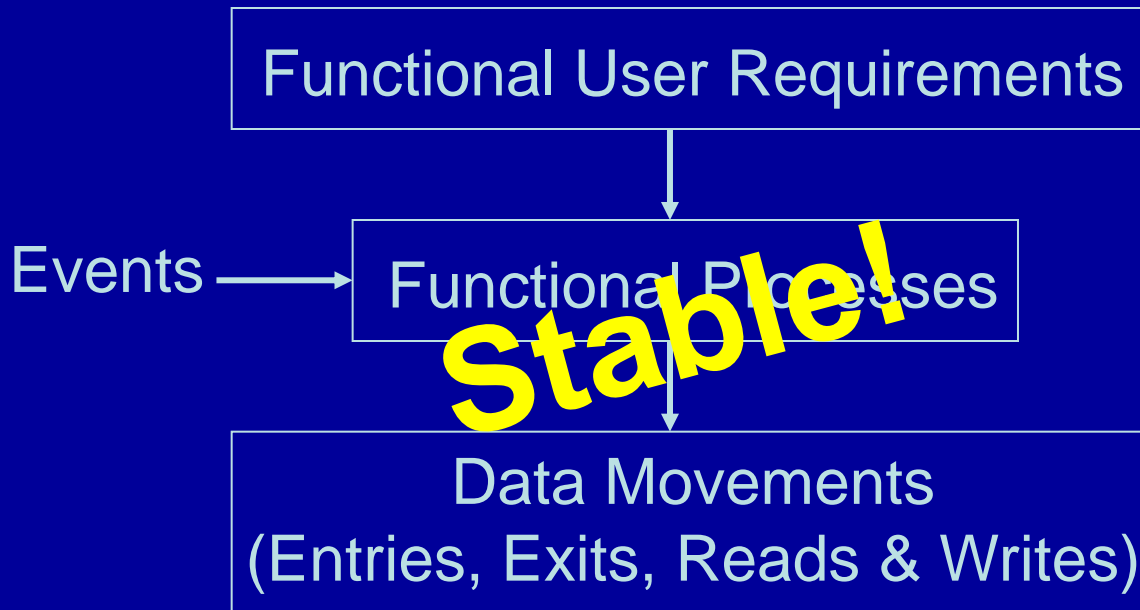
2010

2012

12



The COSMIC Generic Software Model defines the measurement method



Entry = Exit = Read = Write = **1 CFP** – the unit of measure

Size of a Functional Process: Min = 2 CFP; Max = **No limit**



COSMIC size measurements are much more differentiating than IFPUG's

Example: batch processes*

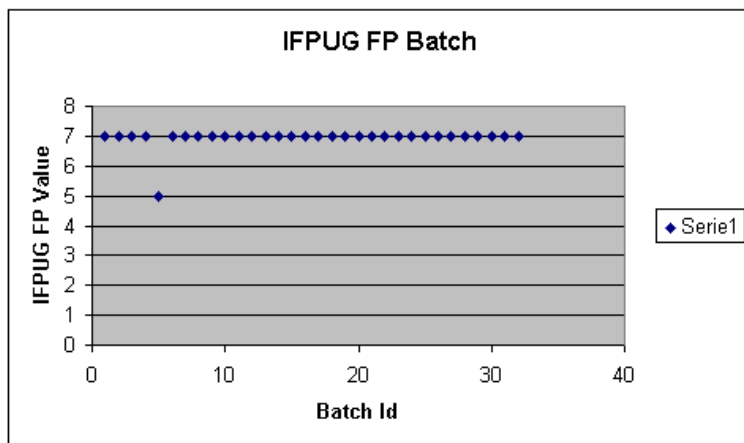


Figure 3 – Distribution of IFPUG FP Batch

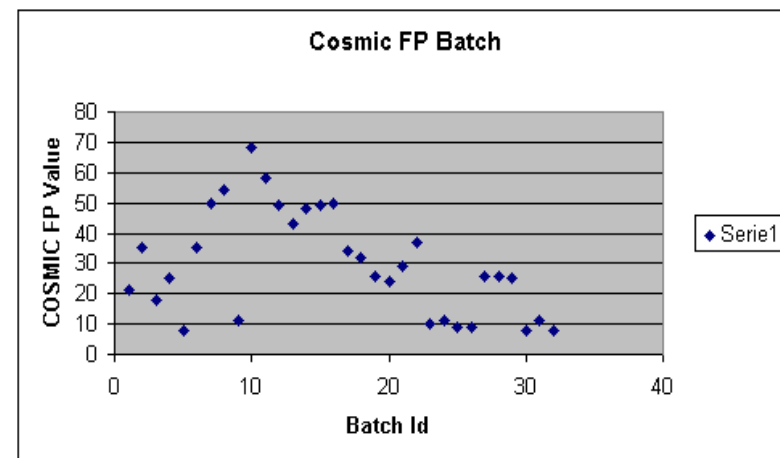
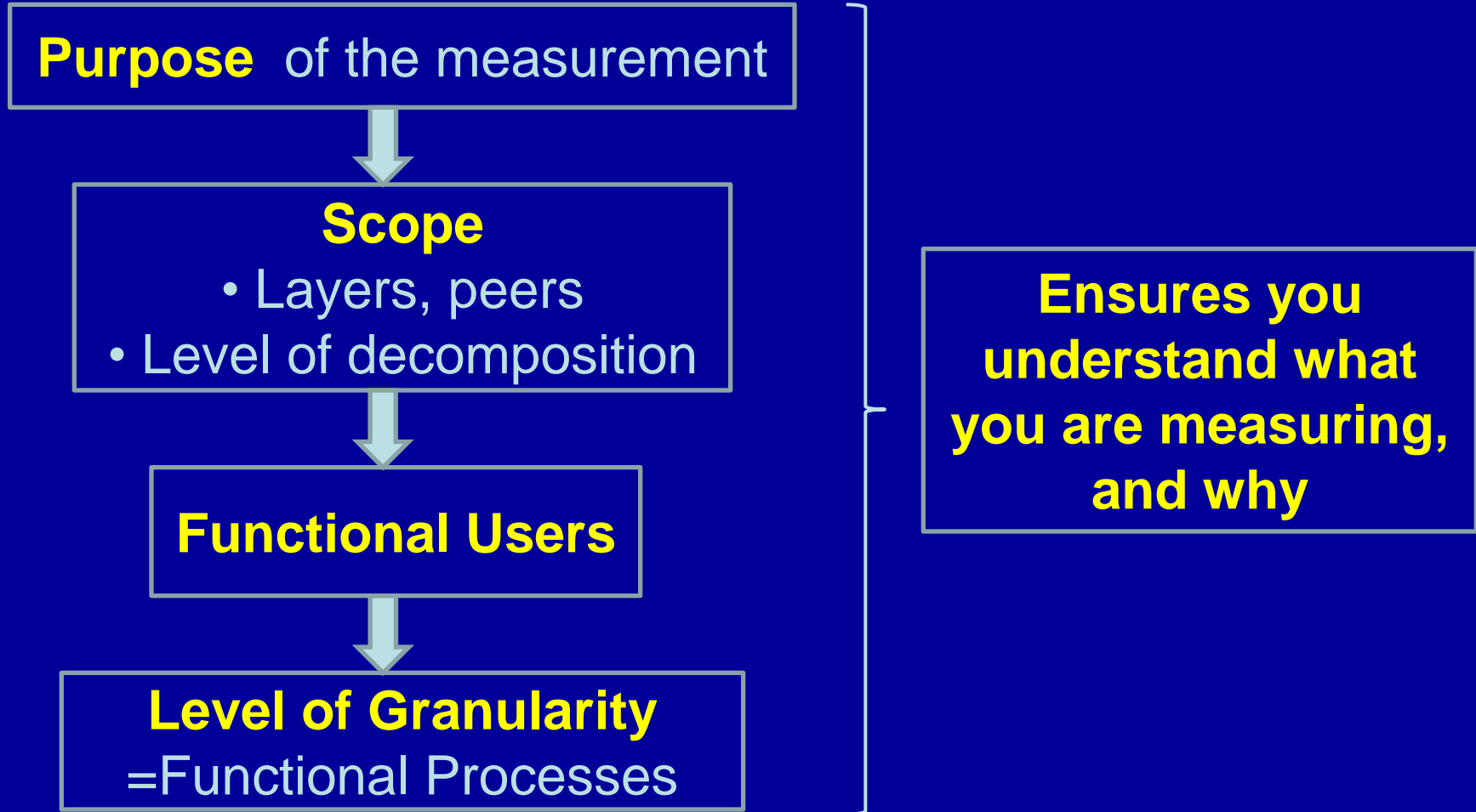


Figure 4 – Distribution of COSMIC FP Batch

**Does this difference in precision matter?
You decide**



Before measuring, we use the ‘Software Context Model’ to determine the ‘Measurement Strategy’





Different functional users see different sizes - examples

Functional User(s)



Whole business application

Human 'end users'



User interface

Business Rules

Data Services

Other components & human 'end users'

Application

SOA component

SOA component

Other SOA components or applications



Mobile phone application

Human 'end users' or hardware devices & peer software?



The Software Context Model needs a little refinement to make it simpler

Purpose of the measurement

Scope

- Level of decomposition
- Layers, Peer items

Functional Users

Level of Granularity
=Functional Processes

Actions for 2011 in discussion

Align definitions with
'software architecture'
standard ISO/IEC 14020

Establish standard
'Viewpoints'



Our support Infrastructure is now very comprehensive

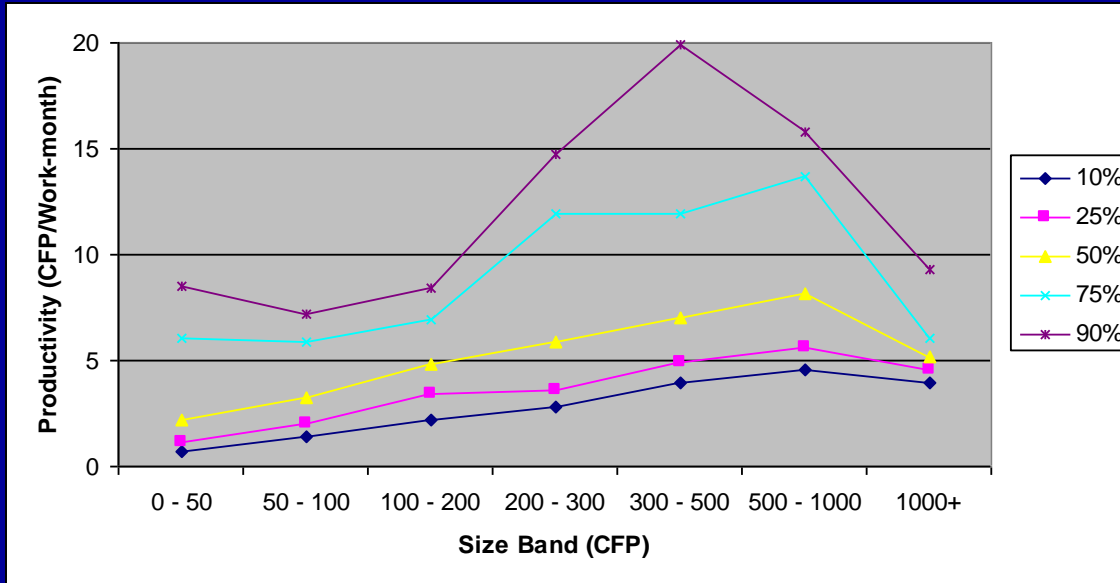
- Basic documentation & translations
- ISO/IEC 19761
- Guidelines
- ISBSG Benchmark data
- Website www.cosmicon.com
- Tools, Case studies
- Certification exams (Entry-level)
- Research spin-off

All COSMIC documentation is available for free download*

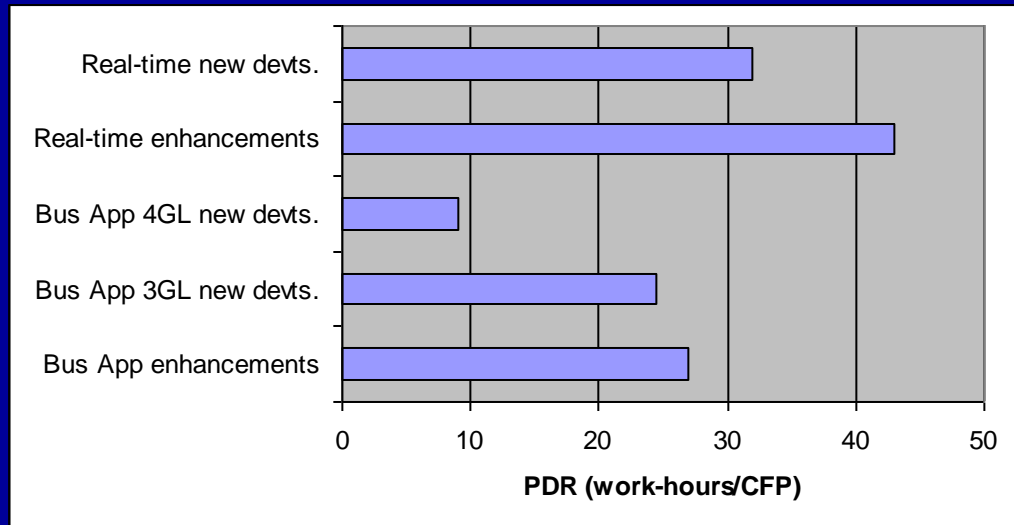
* Except ISO/IEC standards & ISBSG reports



COSMIC/ISBSG benchmark data shows interesting results



'The Performance of Real-Time, Business Application and Component Software Projects', COSMIC/ISBSG, September 2009
www.isbsg.org





We have many activities underway to expand the support infrastructure

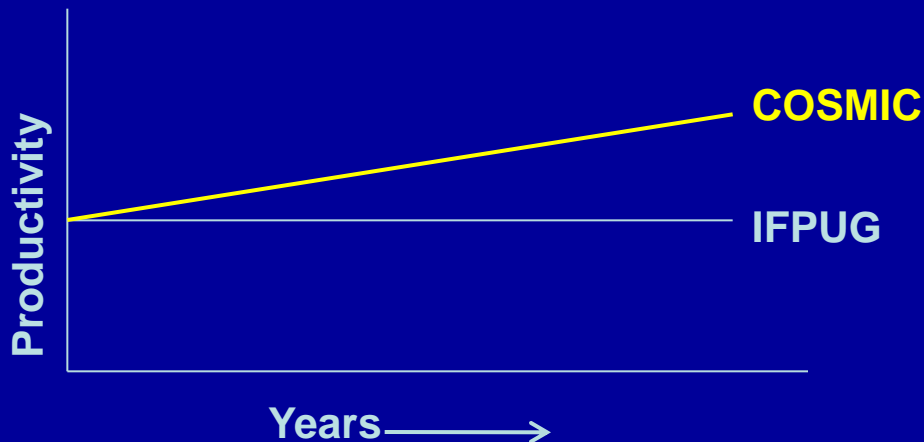
- New Guidelines
 - Sizing real-time applications
 - Sizing in 'agile' projects
 - Project effort estimating
 - Assuring measurement accuracy & repeatability
- New Certification exam (Expert-level)
- Estimating research
- Improved guidance:
 - Convertibility
 - Approximate sizing
- Collect and analyse more benchmark data



Software development productivity improvement revealed

A major European bank

- Invested heavily in process improvement
- Measurements of productivity using IFPUG showed no improvement
- Re-measured some projects using COSMIC
 - **productivity had improved!**



(transactions had become 'more complex' = larger over time)



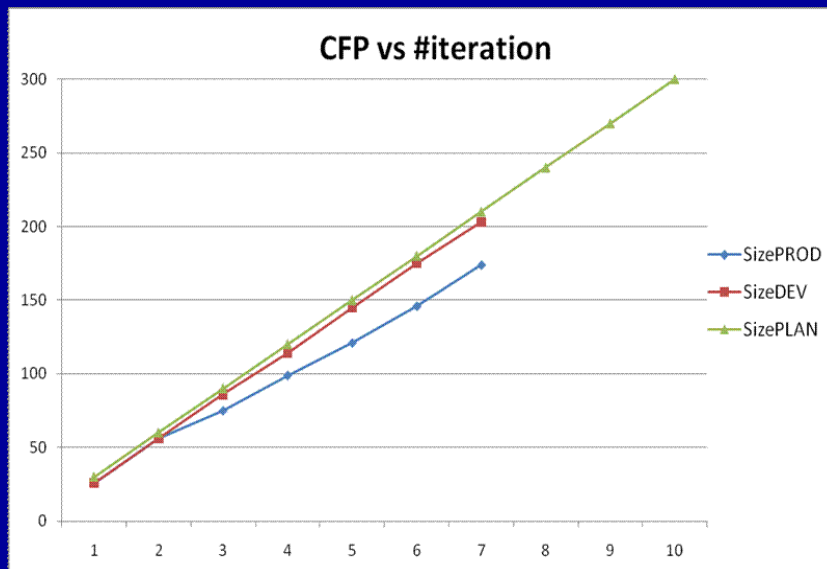
There is a lot of positive experience using COSMIC in 'agile' projects

Story Points

- Subjective measure
- In practice measures 'effort', not 'size'

COSMIC

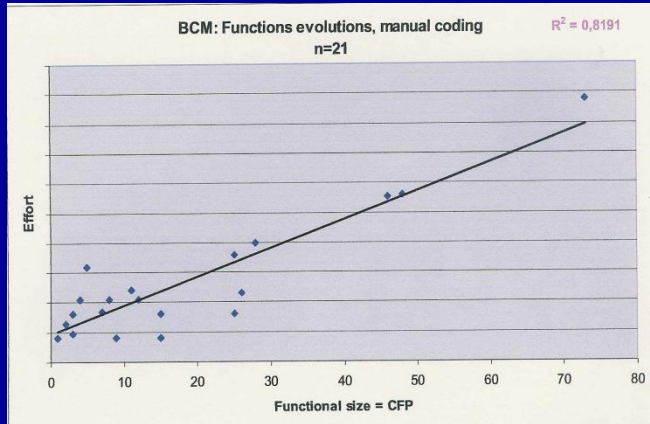
- Objective sizing
- Initial project estimation
- Sprint estimation & project re-estimation
- Process improvement monitoring
- Benchmarking
- Re-work measurement
- EVA



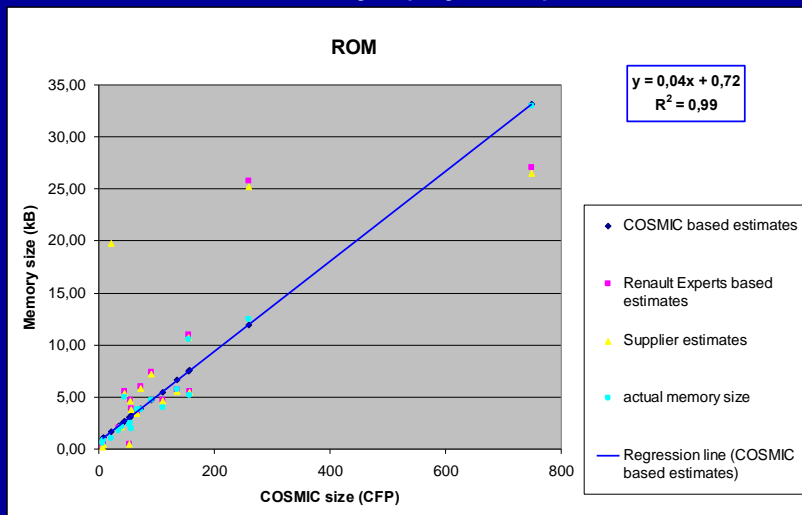


Renault and Saab* estimate development effort and memory sizes for ECU's from CFP

Renault: Devt. effort vs CFP size



Renault: Memory (bytes) vs CFP size



Renault now uses COSMIC FSM to estimate:

- Development effort
- RAM & ROM sizes

Next step:

- Measure sizes automatically from specifications

* Papers available on www.cosmicon.com



The COSMIC method is now used around the world

A selection of users

Finance

- European Union T&CUD
- National Bank of Canada
- Rabobank
- REEAL Insurance

Software Houses

- Atos Origin
- Cognizant Technologies
- CSC India
- Siemens IT S&S
- Sogeti Nederland

Some well-known names

- El Corte Ingles (Spain)
- Foxconn (China)
- Eurocopter (EADS)
- Hitachi (Japan)
- Nokia-Siemens
- NTT-East (Japan)
- Renault, Saab

UK

- HMRC Southend
- Atos Origin
- etc



Conclusions

- COSMIC has achieved its initial goal of developing and bringing to market a more precise and generally-applicable FSM method
- We are now focusing on:
 - expanding the support infrastructure
 - developing COSMIC-based project estimating
 - building on our success stories

Some software customers in industry are now reaping major benefits (and so are some suppliers)



**Thank you for your
attention**

Charles Symons

cr.symons@btinternet.com

www.cosmicon.com