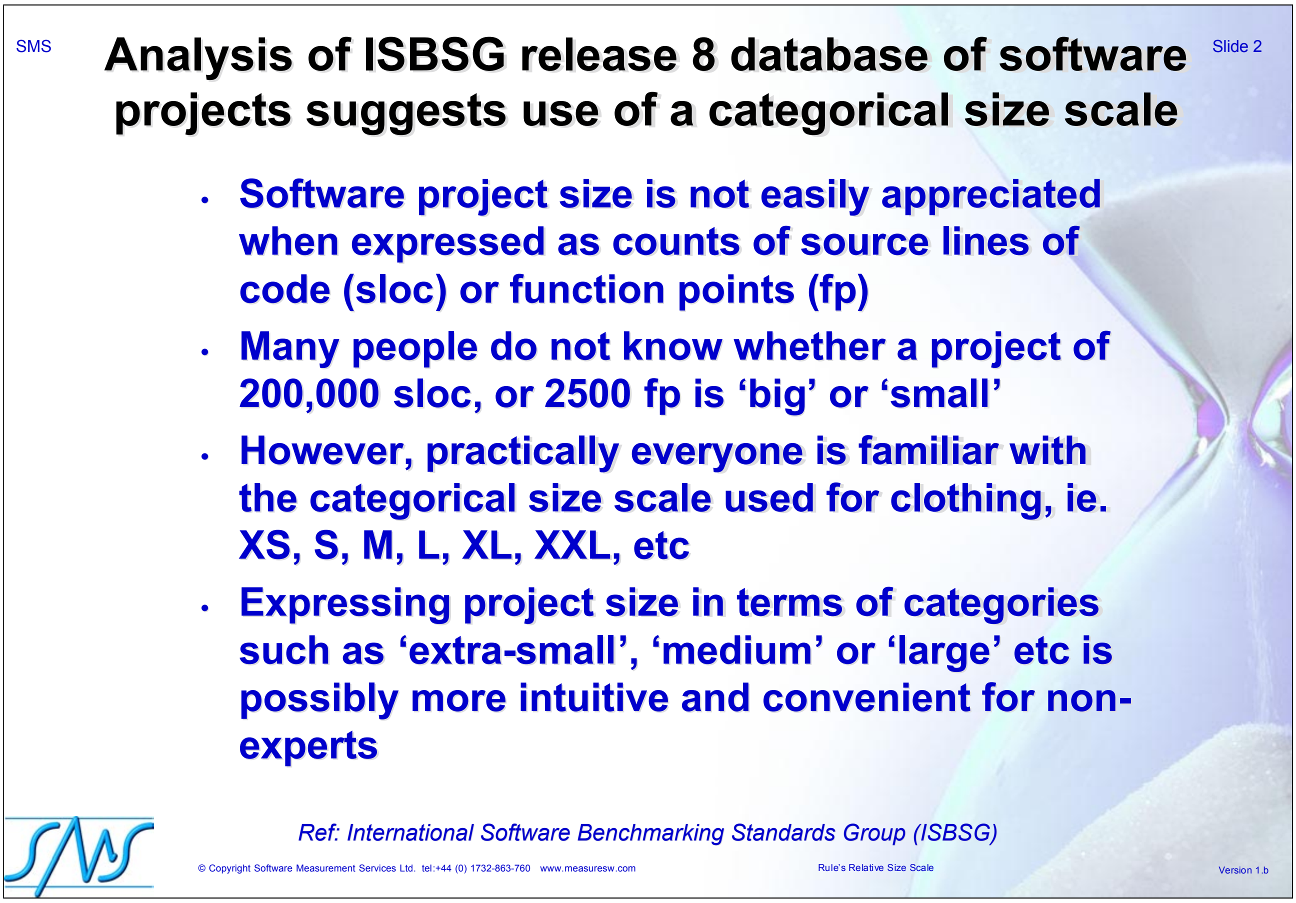
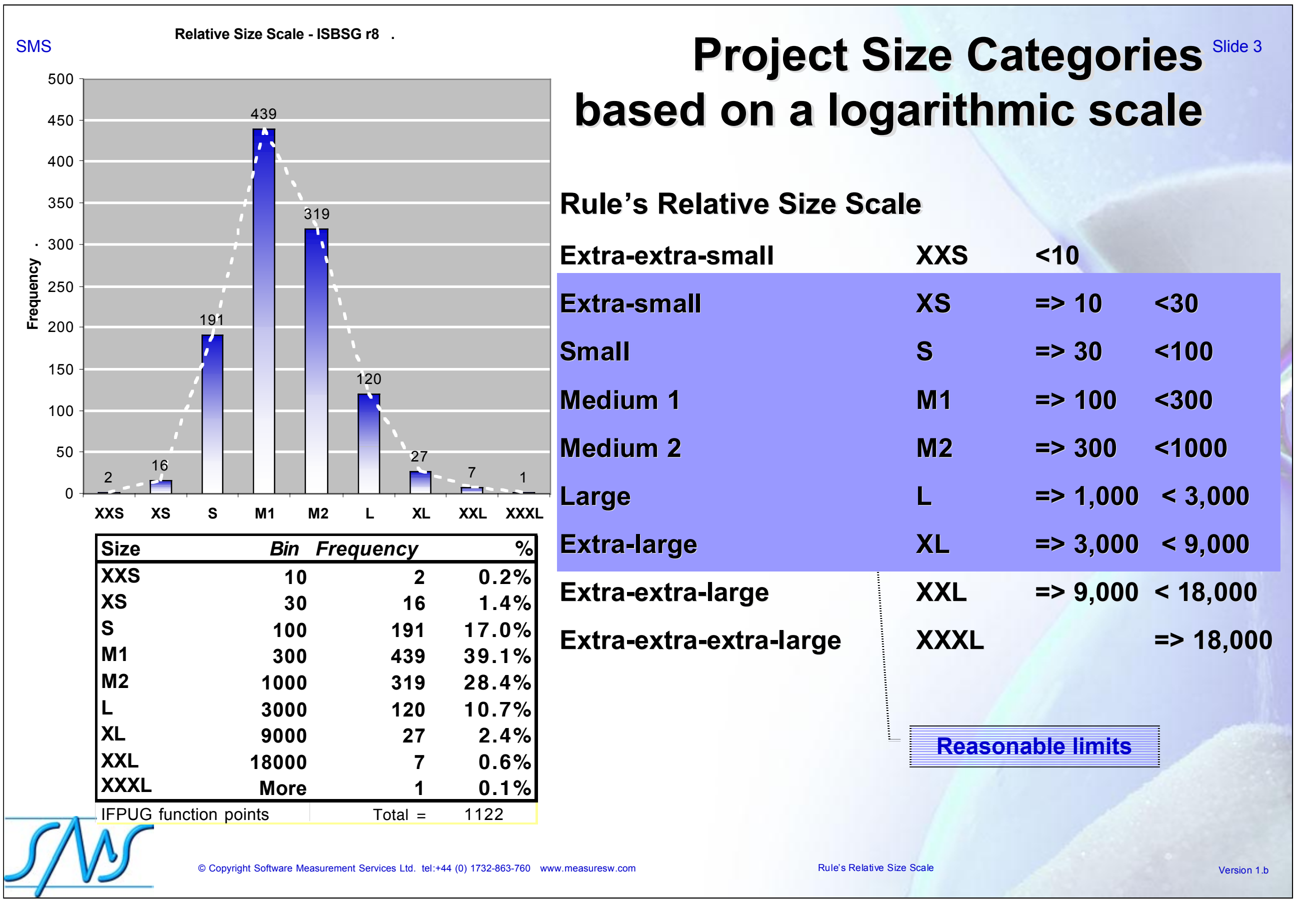

SMS
Slide 1
'Small project',
'medium-size project'
and 'large project':
what do these terms mean?
Software Measurement Services Ltd.
+44(0)1732.863.760
admin@measuresw.com
www.measuresw.com
Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided these copyright and "No Warranty" statements are included with all reproductions and derivative works. See the Notes Page for conditions.
© Copyright Software Measurement Services Ltd. tel:+44 (0) 1732-863-760 www.measuresw.com
Rule's Relative Size Scale
Version 1.b



Rule's Relative Size Scale

Software products come in a wide range of sizes, from trivially small enhancements and individual functional increments that address a single user story, to the massive administrative applications used by national governments and multinational corporations.

For benchmarking and estimating purposes it often is useful to appreciate the relative size of a specific piece of software. An appropriate scale of size categories can be based on the familiar terms used to express clothing sizes i.e. XXS, XS, S, M, L, XL, XXL, XXXL.



Analysing the ISBSG dataset we can construct a set of size category 'bins' that mimics a normal distribution and that demonstrates the central tendency of project size data.

The following table illustrates this categorical scale and indicates the respective function point size (using the IFPUG r4.x standard)

Table : Size categories & their equivalent function point size

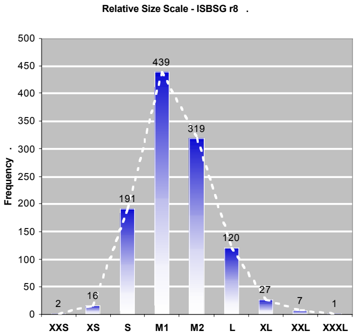
Relative Size	**Size Code**	**Function Point Size (IFPUG)**
Extra-extra-small	XXS	=> 0 and <10
Extra-small	XS	=> 10 and <30
Small	S	=> 30 and <100
Medium1	M1	=> 100 and <300
Medium2	M2	=> 300 and <1000
Large	L	=> 1,000 and < 3,000
Extra-large	XL	=> 3,000 and < 9,000
Extra-extra-large	XXL	=> 9,000 and < 18,000
Extra-extra-extra-large	XXXL	=> 18,000

The 'bin ranges' are adjusted to provide convenient figures. However, the upper limit of each successive size category is more-or-less 3 times that of the next smallest category (except that it seems sensible to reduce this to a factor of 2x for the upper limit of the XXL category.

SMS

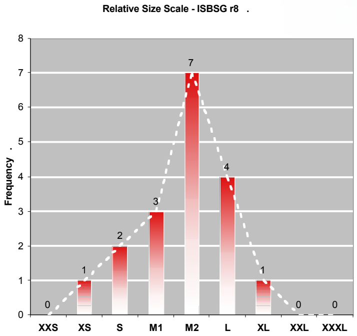
The distribution of project size appears consistent irrespective of the functional sizing method

Slide 4



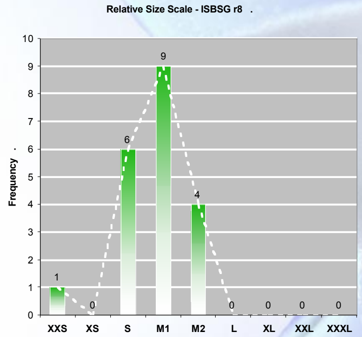
IFPUG

Size	Bin	Frequency	%
XXS	10	2	0.2%
XS	30	16	1.4%
S	100	191	17.0%
M1	300	439	39.1%
M2	1000	319	28.4%
L	3000	120	10.7%
XL	9000	27	2.4%
XXL	18000	7	0.6%
XXXL	More	1	0.1%
IFPUG function points	Total =	1122	



MkII

Size	Bin	Frequency	%
XXS	10	0	0.0%
XS	30	1	5.6%
S	100	2	11.1%
M1	300	3	16.7%
M2	1000	7	38.9%
L	3000	4	22.2%
XL	9000	1	5.6%
XXL	18000	0	0.0%
XXXL	More	0	0.0%
MkII function points	Total =	18	



COSMIC

Size	Bin	Frequency	%
XXS	10	1	5.0%
XS	30	0	0.0%
S	100	6	30.0%
M1	300	9	45.0%
M2	1000	4	20.0%
L	3000	0	0.0%
XL	9000	0	0.0%
XXL	18000	0	0.0%
XXXL	More	0	0.0%
COSMIC full function points	Total =	20	

Non-overlapping datasets from ISBSG r8 - quality rating A & B

 Rule's Relative Size Scale Version 1.b

The ISBSG benchmark dataset actually is composed of three non-overlapping datasets, consisting of projects with their functional size measured respectively using IFPUG FPA, Mk II FPA and COSMIC functional size measures.

Note that each of these three sizing methods is recognised as an ISO standard, compliant with ISO 14143. The respective ISO numbers are:

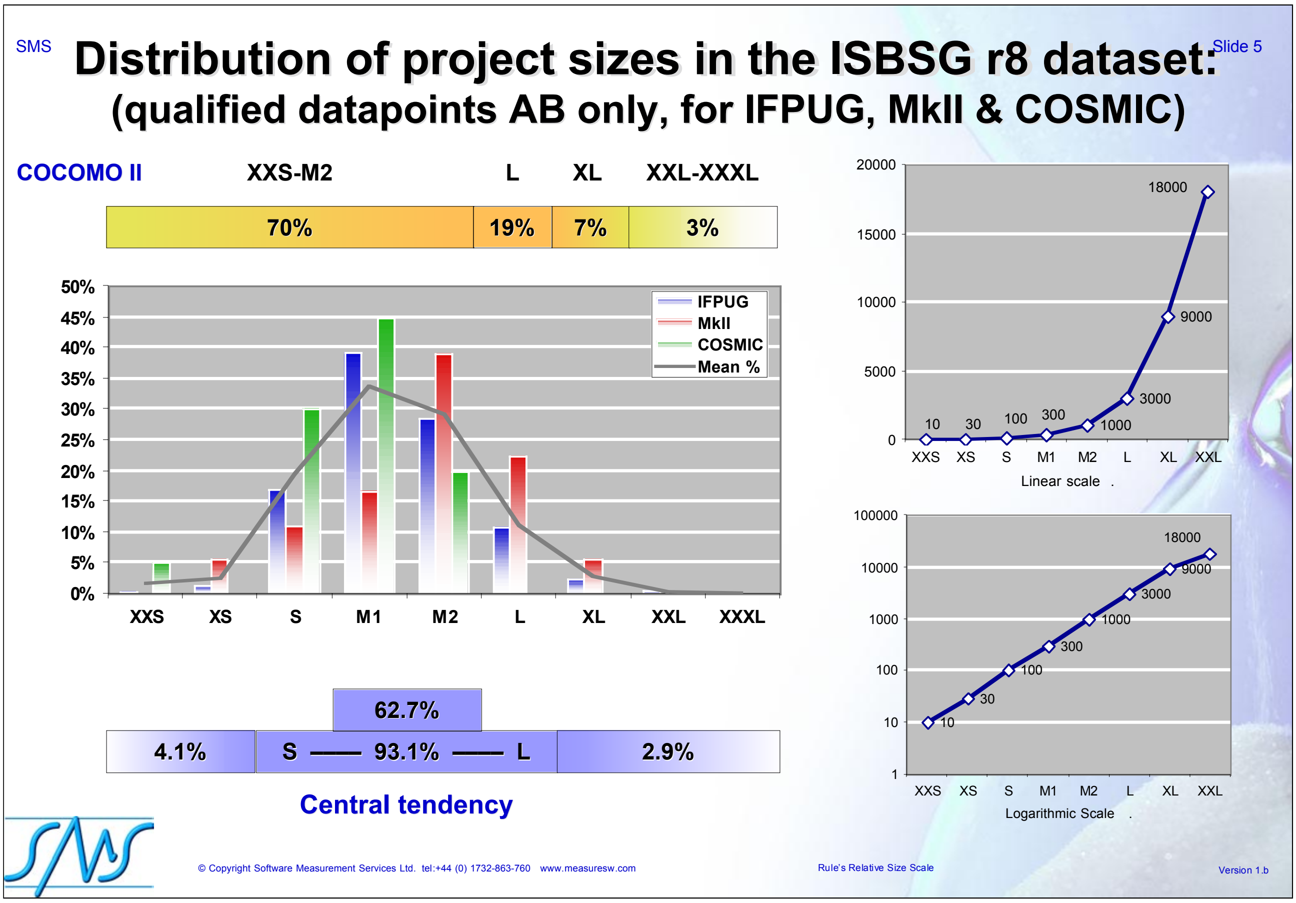
ISO/IEC 20926 = IFPUG FPA release 4.2

ISO/IEC 20968 = Mk II FPA

ISO/IEC 19761 = COSMIC-FFP

Analysing each of these three datasets separately using the size categories suggested gives very similar results.

The results using IFPUG, Mk II and COSMIC-FFP functional size measures are sufficiently similar as to remove any necessity to modify the suggested ‘bin ranges’. A single consistent categorical scale can be used irrespective of the functional size measurement scale preferred by the user.



The vast majority of projects (>93%) fall into the Small to Large size range. That is, in the range =>30 to <3,000 function points).

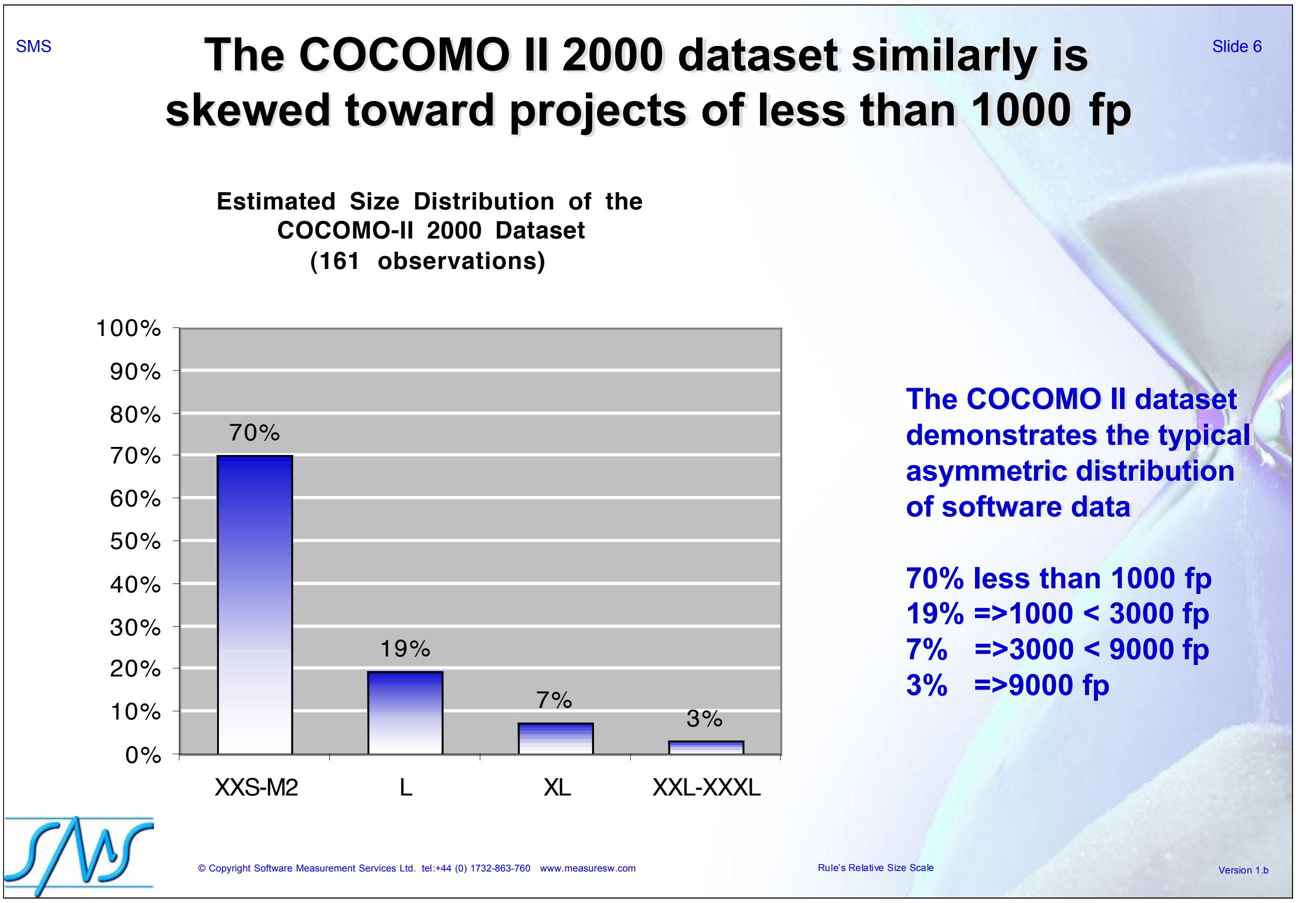
Projects that produce fewer than 30 function points usually are incremental enhancement projects (note that corrective maintenance, ie. ‘bug fixes’, to make the product behave as advertised, deliver no change in the size of the functional requirements, of course).

Projects of over 3000 function points represent less than 3% of projects.

Extra-Large category projects that deliver between 3000 and 9000 function points usually take years rather than months and involve teams composed of tens or hundreds of software developers. Telephone billing systems, stock control applications and accounting packages may fall into this XL category.

There are very few XXL and XXXL projects delivering more that 9000 function points. As the population of such projects is small, it is difficult to draw statistically valid conclusions about projects of this size, or to contrive cost models that accurately predict project performance. As such projects employ hundreds of developers over periods of years, they develop their own, unique environments and cost models.

Strong evidence suggests that the risk of project failure increases with size. Many if not the majority of projects in the XL to XXXL size range fail to fulfil their user requirements and never deliver the business benefits that supposedly justify their existence. Many are abandoned prematurely due to budget & schedule over-runs. Beware of becoming involved in such projects.



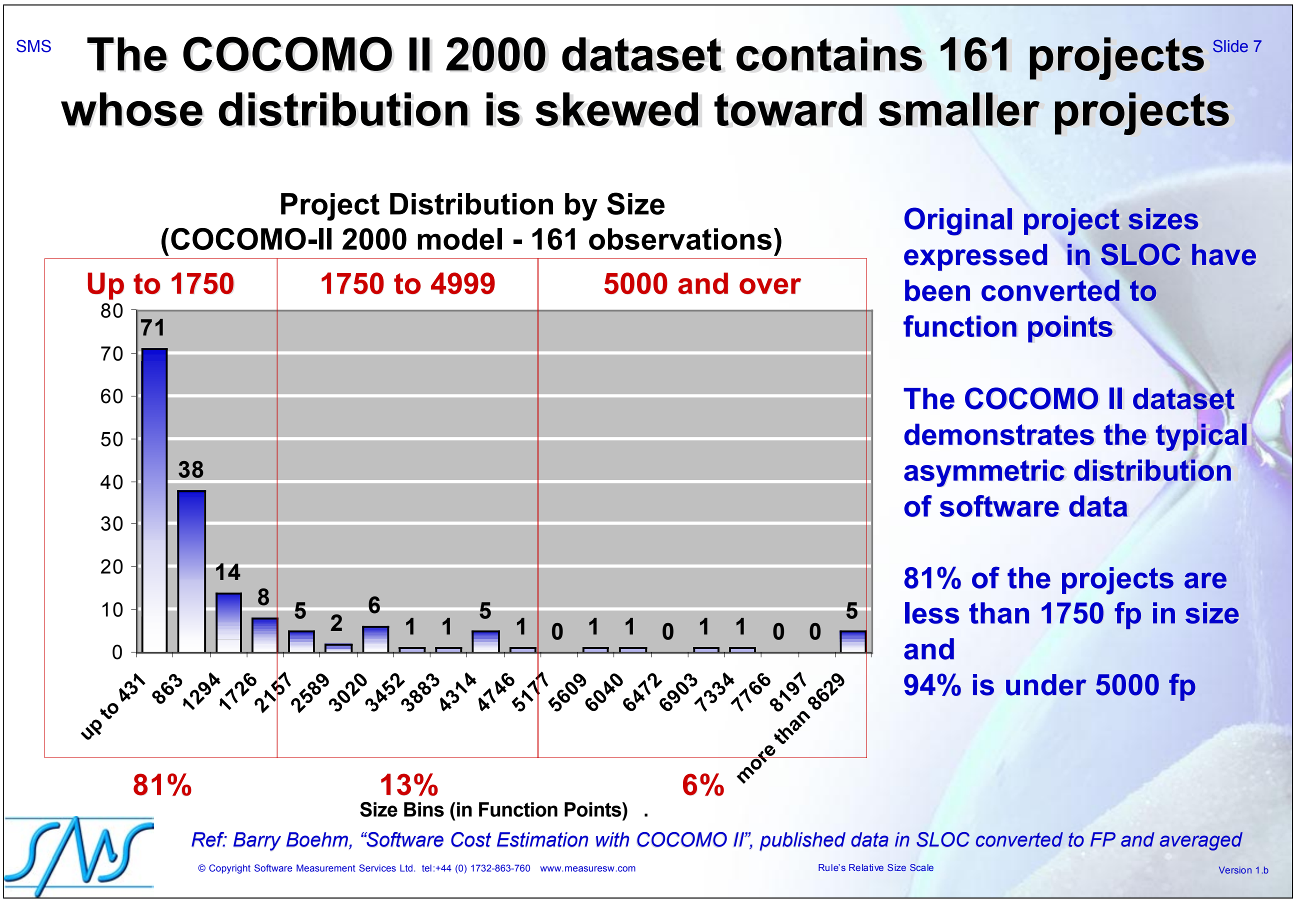
To provide a 'sanity check' on the analysis of the ISBSG datasets, a similar analysis was performed for the dataset from which the COCOMO II 2000 cost model has been derived.

This analysis is broadly in agreement with the ISBSG analysis. That is, some 89% of projects fall into the XXS to L size range. (Note that, unfortunately, the available published data does not enable a distinction to be made between projects with a size of less than about 500fp).

About 7% of projects fall in the XL category (ie. =>3000 and <9000 fp).

Only 3% fall into the XXL to XXXL categories (ie. =>3000 fp upward).

While this suggests that a higher proportion of the COCOMO II 2000 projects are 'larger' projects than in the ISBSG datasets, this is entirely consistent with the nature and purpose of the COCOMO cost model, which primarily is aimed at large-scale development programmes. One would expect the COCOMO II 2000 dataset to contain bigger projects.



In fact, 94% of projects in the COCOMO II 2000 dataset are under 5000 fp in size and the dataset is highly skewed towards smaller projects.

This is consistent with other observations.

Analysis of the COCOMO II 2000 dataset provides results that are broadly compatible with the categorical scale resulting from analysis of the ISBSG dataset.

SMS

The Relative Size Scale can be combined with productivity benchmark data to estimate effort & cost

Slide 8

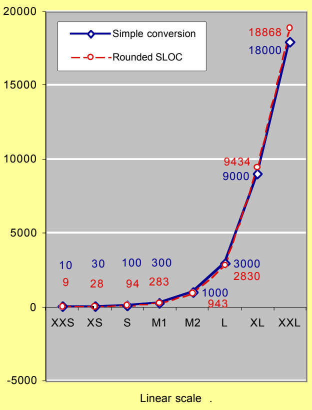
Expressing Rule's Relative Size Scale in Source Lines Of Code
Assume C++ or Java language, converting at 53 SLOC/FP

Simple conversion		
	Bin Upper Limit	SLOC/FP = 53
Size	FP	SLOC (C++ or Java)
XXS	10	530
XS	30	1,590
S	100	5,300
M1	300	15,900
M2	1000	53,000
L	3000	159,000
XL	9000	477,000
XXL	18000	954,000
XXXL	More	More

Rounding the SLOC ranges to aid memorability gives...

Rounded SLOC			
FP	SLOC	variance fp	variance %
9	500	-1	-6%
28	1,500	-2	-6%
94	5,000	-6	-6%
283	15,000	-17	-6%
943	50,000	-57	-6%
2830	150,000	-170	-6%
9434	500,000	434	5%
18868	1,000,000	868	5%
More	More		

The graph shows using rounded SLOC counts has trivial impact



Baseline estimates using Rule's Relative Size Scale
Assume C++ or Java, median productivity (ref: ISBSG r8)

New Development		assume efficiency of 80%	
wh/fp =	15	wh/FTE/e.day	6.4
Size	FP	wh	FTE days
XXS	10	150	23
XS	30	450	70
S	100	1,500	234
M1	300	4,500	703
M2	1000	15,000	2,344
L	3000	45,000	7,031
XL	9000	135,000	21,094
XXL	18000	270,000	42,188
XXXL	More	More	More

Incl. accommodation, heat, light, etc

Fully burdended costs	
FTE/day=	£ 300
Size	Cost
XXS	£ 7,031
XS	£ 21,094
S	£ 70,313
M1	£ 210,938
M2	£ 703,125
L	£ 2,109,375
XL	£ 6,328,125
XXL	£ 12,656,250
XXXL	More

Enhancement		assume efficiency of 80%	
wh/fp =	34	wh/FTE/e.day	6.4
Size	FP	wh	FTE days
XXS	10	340	53
XS	30	1,020	159
S	100	3,400	531
M1	300	10,200	1,594
M2	1000	34,000	5,313
L	3000	102,000	15,938
XL	9000	306,000	47,813
XXL	18000	612,000	95,625
XXXL	More	More	More

Fully burdended costs	
FTE/day=	£ 300
Size	Cost
XXS	£ 15,938
XS	£ 47,813
S	£ 159,375
M1	£ 478,125
M2	£ 1,593,750
L	£ 4,781,250
XL	£ 14,343,750
XXL	£ 28,687,500
XXXL	More

Rule's Relative Size Scale / **Baseline Estimates** (assume efficiency of 80%)

Bin Upper Limit		SLOC/FP = 53	New Development		Enhancement	
		C++ or Java	wh/fp =	15	wh/fp =	34
Size	FP	SLOC	wh	FTE days	wh	FTE days
XXS	10	500	150	23	340	53
XS	30	1,500	450	70	1,020	159
S	100	5,000	1,500	234	3,400	531
M1	300	15,000	4,500	703	10,200	1,594
M2	1000	50,000	15,000	2,344	34,000	5,313
L	3000	150,000	45,000	7,031	102,000	15,938
XL	9000	500,000	135,000	21,094	306,000	47,813
XXL	18000	1,000,000	270,000	42,188	612,000	95,625
XXXL	More	More	More	More	More	More

Fully burdended costs

FTE/day =	£ 300
New Development	Enhancement
GBP	GBP
£ 7,031	£ 15,938
£ 21,094	£ 47,813
£ 70,313	£ 159,375
£ 210,938	£ 478,125
£ 703,125	£ 1,593,750
£ 2,109,375	£ 4,781,250
£ 6,328,125	£ 14,343,750
£ 12,656,250	£ 28,687,500
More	More

Assumptions & abbreviations

assumes software language is C++ or Java
assumes 'typical' conversion of FP to SLOC
assumes median work hours per FP
assume efficiency = 80% (ie. 6.5 wh/staff day)
assumes no (dis)economies of scale
assumes full resourcing
assumes no overhead & no wasted capacity
FP = Function Points
SLOC = Source Line Of Code
wh = work hour
FTE = Full Time Equivalent
GBP = Great British Pounds (sterling)

The Relative Size Scale can be expressed using a conversion factor of 53 SLOC/FP for the popular Java & C++ languages, and by approximating the bin ranges for convenience & memorability. This may help those project managers unfamiliar with function point analysis. Organisations using other programming languages may derive alternative 'bin ranges' based on appropriate SLOC/FP conversion factors.

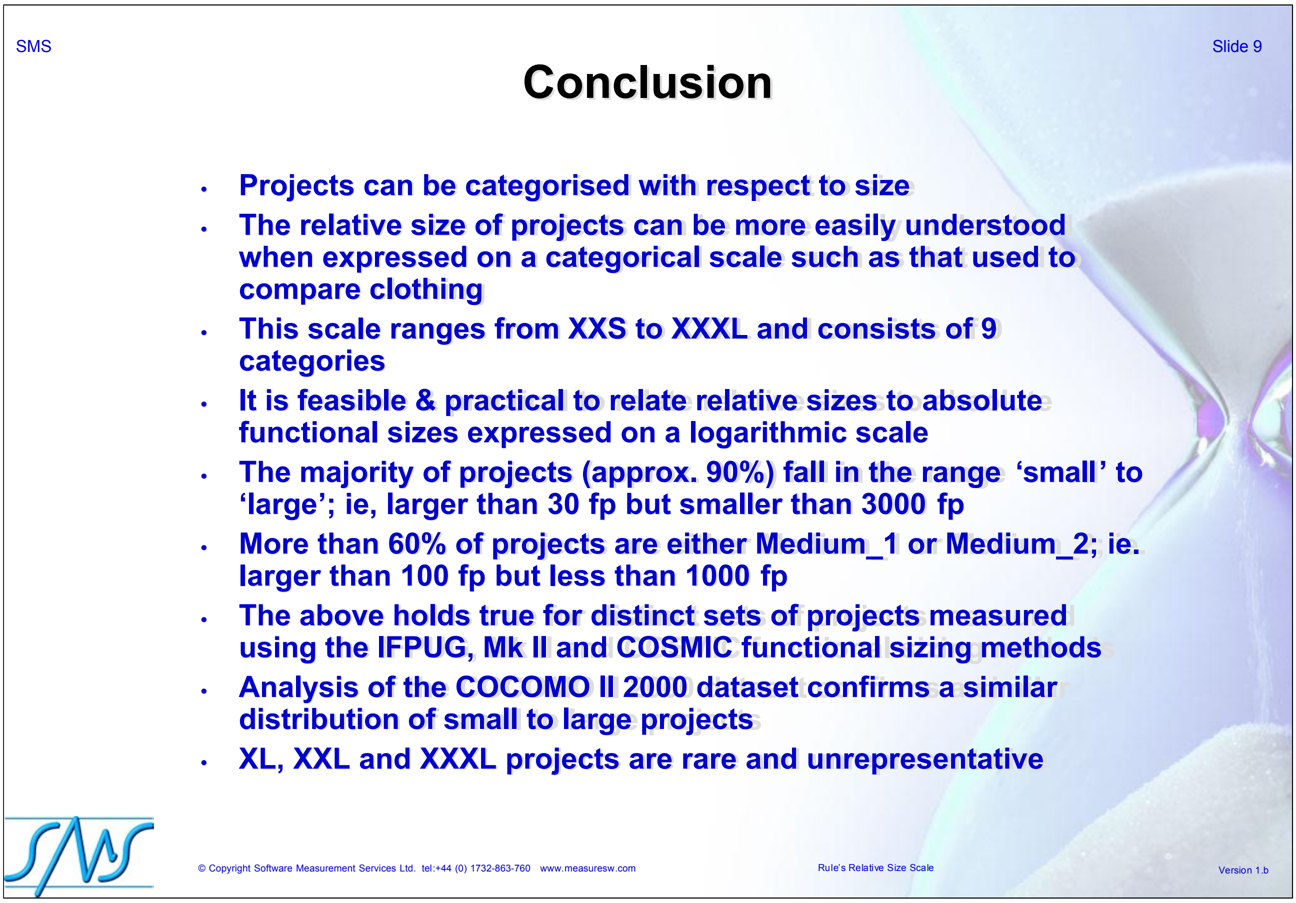
Furthermore, using benchmark productivity & unit cost figures for New Development and Enhancement projects, we can derive from the size categories estimates of effort & cost.

The cost figures quoted are in Pounds Sterling (GBP) valued as at 1st January 2004. For projects performed outside the United Kingdom,such as those using outsourced resources resident offshore, the Full Time Equivalent rate should be adjusted to reflect the fully burdened cost of employing staff in the pertinent environment.

The productivity figure used represents the ***median*** performance of software projects (the distribution of project productivity results is very wide and skewed toward the low end of the scale). Productivity may be higher (or lower!) in your project. However, as the majority of organisations developing and/or enhancing software-intensive systems exhibit low-maturity, the use of more optimistic figures needs to be justified by objective evidence and local data.

High-maturity organisations with better productivity and lower unit costs may use work-hour/function point and cost/FTE-day factors more in keeping with their local experiences.

Note that the effort & cost estimates derived cover the software development life-cycle (SDLC) from the start of requirements capture through to the end of user acceptance testing. They incorporate allowances for the effort contributions made by supply-side staff (ie. the project manager, project administration & support staff, developers, testers and quality assurance staff). However, they ***do not*** incorporate any allowance for user training or system roll-out, nor the effort contributed by customer-side stakeholders. No assumptions of any (dis)economies of scale or allowance for other influences on performance have been made.



- Data from four non-overlapping datasets of software projects provide compatible results when categorised into 9 size categories. These can be compared to functional sizes measured on a logarithmic scale (the upper limit of each category is approximately 3 times the upper limit of the preceding category).
- For the purposes of comparing the size category of projects, there is little to distinguish between any of the three standard variants of functional size measurement.
- From the ISBSG data, more than 93% of projects are distributed over the Small, Medium_1, Medium_2, to Large categories. That is, 93% of reported projects fall into the range =>30 and <3000 function points.
- Comparison with the COCOMO II 2000 dataset suggests that, while there may be a wider spread of projects toward the 'larger' sizes than is suggested by the ISBSG data, project size nevertheless is expected to be highly skewed toward the 'small' end of the scale.
- 89% of projects in the COCOMO II 2000 dataset fall into the XXS to L categories.
- Hence, for the purpose of estimating, it is suitable to propose the following heuristics:
    1. Functional size of software projects may be distributed normally across 9 size categories (ie. respectively, up to 10, 30, 100, 300, 1000, 3000, 9000, 18000, and more than 18000 function points);
    2. 90% of software projects fall into the Small-Large size range (ie. =>30 <3000 fp);
    3. 60% of software projects fall into the M1-M2 range (ie. =>100 <1000 fp)
    4. Projects in the XL to XXXL range (>=9000 fp) are rare and unwise. Little is known about the viability and performance of such large projects as few are reported to end successfully by delivering value and the expected benefits to the customer.

Glossary Of Terms

COCOMO II 2000 = The Constructive Cost Modelling technique

A software cost model originated by Prof. Barry Boehm in the 1980’s. COCOMO is a widely-recognised model for estimating effort and cost based on product size and a number of size and cost drivers: the technique has been updated to give COCOMO II 2000 as the latest version: the design authority is the University of Southern California (http://sunset.usc.edu/research/COCOMOII/)

COSMIC =COmmon Software Measurement International Consortium

An international consortium of software measurement experts from over 19 countries and from both government, commerce and academia set up in 1997 expressly to create improved measures for software systems (http://www.cosmicon.com/

FP = function point

A unit of measurement applied to the ‘amount of functionality required by or delivered to the external users of a system’

FPA = function point analysis

A technique for measuring the functional size of a system

FSM = functional size measure

The generic term for that set of techniques used for measuring the amount of functionality required by or delivered to the external users of a system that comply with ISO 14143

FTE = Full Time Equivalent

A measure of effort amounting to one staff member working full time

IFPUG = International Function Point Users Group

The design authority for IFPUG FPA, the technique most closely based on Allan Albrecht’s concepts originated in 1977-82 (http://www.ifpug.org/)

ISBSG = International Software Benchmarking Standards Group

An not-for-profit organisation that collects, analyses and makes available software project benchmark data on behalf of a number of national software measurement associations (http://www.isbsg.org.au/)

ISO = International Standards Organisation

The design authority for the well-known set of standards adopted by the majority of countries and recognised by the appropriate national standards bodies for a wide range of subjects

SDLC = software (or system) development life-cycle

The set of work products and the activities performed to produce them, usually organised into sequences of tasks, and grouped into steps and stages that lead to creation of a project’s deliverables. SDLC’s may take many forms. Examples are: waterfall, V-model, spiral, incremental, lean & agile, etc.

SLOC = source line of code

One line of the code written by a computer programmer - such code may be written in any one of a variety of programming languages, of varying ‘power’(eg. 1st, 2nd, 3rd, 4th or 5th generation)

UKSMA = United Kingdom Software Metrics Association

The design authority for Mk II FPA, the technique adopted as the UK Government’s preferred measure of system size, originated by Charles Symons in 1984 as an improvement upon Allan Albrecht’s ideas (http://www.uksma.co.uk/)