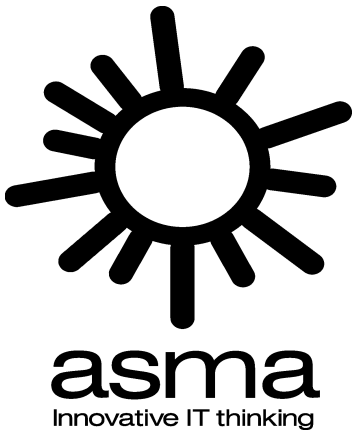

Metrics Matters

The Australian Journal of Software Metrics

December 2001

CONTENTS

About Metrics Matters

The Metrics Matters journal is a publication of the Australian Software Metrics Association and is circulated to ASMA members Australia wide.

CONTRIBUTIONS

Metrics Matters is seeking contributions from IT professionals. If you have any form of contribution, such as:

- Case studies
- Seminar notes and write-ups
- Research work
- Industry surveys
- Current practices
- Methods and techniques

Please email them to the publisher on **asma@cia.com.au** or send them on 3.5" floppy disk or hardcopy to the address below. The document must be in Rich Text Format (.rtf).

The Publisher, Metrics Matters
ASMA (NSW)
PO Box 432
Leichhardt. NSW. 2040

ADVERTISEMENTS

If you wish to advertise your products or services in Metrics Matters and reach metrics-aware individuals and companies, please contact the Publisher.

Publisher
Maria Patane`

Chief Editor
Rocco Muller

Disclaimer

The opinions expressed here are those of the individual authors and are not necessarily those of the Editor or the Australian Software Metrics Association.

Software Metrics Articles

The new ISBSG Estimating, Benchmarking and Research Suite

By
Peter Hill
ISBSG

The International Software Benchmarking Standards Group has just released "The ISBSG Estimating, Benchmarking and Research Suite", (incorporating the Data Disk Release 7). The major components of this CD ROM based product are a flexible project estimation tool, (the "Reality Checker") and data on over 1,200 completed software projects.

This powerful suite of tools and data can be used by:

- Project Managers and anyone involved in software estimation.
  The estimation tool provides estimates based on regression analysis of the ISBSG data, while the data itself can be used for comparison and analogy based estimation.
- Users of commercial estimation tools.
  Some commercial estimation tools, like "Predictor" provide for the use of the ISBSG data contained on the CD.
- I.T. Planners for development analysis.
  The ISBSG data can be used to analyse and compare the performance of languages, tools, methods and techniques. This allows decisions to be made about the most appropriate hardware/software platforms for a project or development environment.
- Software Metrics Consultants.

The ISBSG data allows consultants to provide their customers with benchmarking services, estimation information and guidance on the factors that impact development productivity.

- *Standards Assessment, Process Improvement and Quality Assurance, (CMM, ISO), Consultants. The ISBSG data provides consultants with a base for their client assessments plus valuable benchmarking data.*
- Outsourcing Managers.
  The data provides a benchmark that can be used as a base for outsourcing contracts and performance agreements.
- Submitter of projects.
  Organisations that have submitted projects to the ISBSG Repository can benchmark their projects against like projects and can analyse their "repository" within the ISBSG data.
- Researchers.
  Academics can make use of the ISBSG data in their research work.

The Reality Checker is a macro-estimation tool that uses the ISBSG project data to provide fast, interactive project estimates based on the user's choice of variables: project size, team size, platform, language type etc. These variables can be changed "on the fly" to reveal the impact of changing team size, for example. The tool also provides a confidence percentage based on the size and quality of the project data sample that it used in the calculation. This is an ideal way to check your micro estimate.

You can try out a sample of the tool at:
http://www.isbsg.org.au/reality.htm

The ISBSG Estimation, Benchmarking and Research Suite CD is available from ASMA or the ISBSG web site.

….end….

Web site of the month

Readers interested in software metrics and associated software development topics may find the following university related web site interesting.

www.lrgl.uqam.ca/cosmic-ffp/

It is maintained by the Software Engineering Management Research Laboratory at the Quebec University at Montreal, and contains case studies, publications and the COSMIC Measurement Manual Version 2.1- which explains FFP. It also contains links to Software Measurement Organisations.

… end …

Uses and Benefits of Function Points: Parts 2 and 3

Software Construction and Post Implementation

By Pam Morris (Total Metrics)
Email :Pam.Morris@Totalmetrics.com (03 9882 7611)

INTRODUCTION

This is the second and third parts of a four part series to be published over three issues of Metrics Matters. Each section focuses on a different part of the software development life cycle with respect to the different:

- ***benefits*** that can be achieved by an organisation that collects function point measures.
- ***uses*** for function point analysis beyond simply measuring productivity and improving project estimates.

The series looks at the ways function points can assist in better management and control of software development projects in:

Part 1. ***Planning***
Part 2. ***Construction***
Part 3. ***Post-Implementation***
Part 4. ***Customising packages***

FPA Uses and Benefits in ***Project Construction***

Monitoring Functional Creep

Function point analysis provides project management with an objective tool by which project *size* can be monitored for change, over the project's lifecycle.

As new functions are identified, functions are removed or changed during the project the function point count is updated and the impacted functions appropriately flagged. The [1]project scope can be easily tracked and reported at each of the major milestones.

[1] The Victorian State Government in Australia has adopted a recommended policy for Government departments to manage and control government out-sourced development projects using Function Points. Suppliers tender for the development based on a fixed price in dollars per function point. Scope changes are automatically charged by the supplier at a pre-determined contracted charge-rate based on the number of function points impacted and the stage at the life cycle when the change was introduced. The government policy underpinning this approach is called 'Southern Scope'. More information is available at: www.mmv.vic.gov.au/southernscope

If the project size exceeds the limits allowed in the initial estimates then this will provide an early warning that new estimates may be necessary or alternatively highlight a need to review the functionality to be delivered by this release.

Assessing and Prioritizing Rework

Function Point Analysis allows the project manager to objectively and quantitatively measure the scope of impact of a change request and estimate the resulting impact on project schedule and costs. This immediate feedback to the user on the impact of the rework allows them to evaluate and prioritise change requests.

The cost of rework is often hidden in the overall project costs and users and developers have no means to quantify its impact on the overall project productivity rates. Function point analysis enables the project manager to measure the functions that have been reworked due to user-initiated change requests. The results provide valuable feedback to the business on the potential cost savings of committing user resources early in the project to establish an agreed set of requirements and minimising change during the project lifecycle.

FPA Uses and Benefits after ***Software Implementation***

Planning Support Resources and Budgets

The number of personnel required to [2]maintain and support an application is strongly related to the application's size. Knowing the functional size of the application's portfolio allows management to confidently budget for the deployment of support resources. The following figure demonstrates this relationship as demonstrated within an Australian financial organisation. The average maintenance assignment scope (number of function points supported per person) for this organisation is 833 function points per person. The assignment scope has been found to be negatively influenced by the age of the application and the number of users i.e. as both these parameters increase the assignment scope decreases. [3]Capers Jones figures show similar assignment scopes where for ageing, unstructured applications with high complexity an assignment scope of 500 function points per person is not unusual whereas newer, structured applications, skilled staff can support around 1500 – 2000 function points.

[2] Where maintenance and support includes defect repairs and very minor enhancements.

[3] Capers Jones – Applied Software Measurement – Assuring Productivity and Quality – McGraw Hill – Software Engineering series 1991 – Chapter 3 Page180.

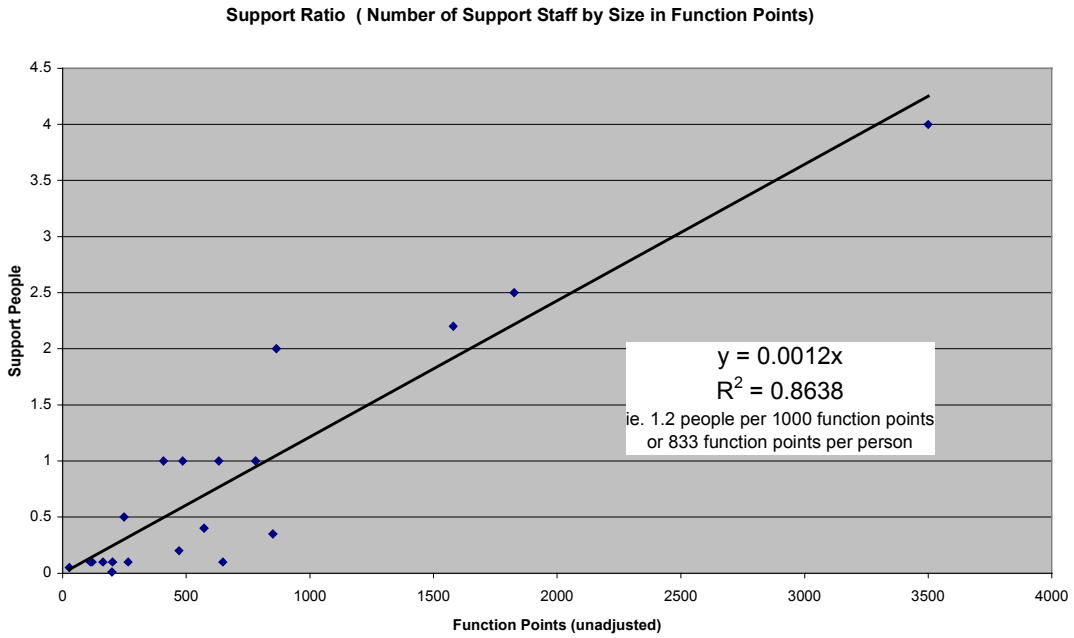

Figure 1 Relationship between the Size of an Application and the Number of Support staff (Source - Total Metrics 1999)

Once implemented, applications typically need constant enhancement in order to respond to changes in direction of an organisation's business activities. Function points can be used to estimate the impact of these enhancements. The baseline function point count of the existing application will facilitate these estimates. As the application size grows with time the increasing assignment scope will provide the justification to assign more support staff.

Benchmarking

The function point count of delivered functionality provides input into productivity and quality performance indicators. These can then be compared to those of other in-house development teams and implementation environments. Benchmarking internally and externally with industry data enables identification of best practice. External benchmarking data is readily available in the ISBSG[4] Repository.

Identifying Best Practice

Project managers seeking 'best practice' in their software development and support areas recognise the need to adopt new tools, techniques and technologies to improve the productivity of the process and quality of the products they produce. Baselining current practice enables management to establish current status and set realistic targets for improvement. Ongoing measurement of productivity and quality key performance

[4] For information on how to access the ISBSG data visit : www.ISBSG.org.au

indicators enable management to assess the impact of their implemented changes and identify where further improvements can be made. Function points are the most universally accepted method to measure the output from the software process. They are a key metric within any process improvement program because of their ability to normalise data from various software development environments combined with their ability to measure output from a business perspective as compared to a technical perspective.

Planning New Releases

The functional hierarchy of the functionality delivered by an application can also assist the support manager in planning and grouping change requests for each new release of the application. The hierarchy illustrates closely related functions and their relative size. If the impact of change is focused on a group of related functions then development effort will be reduced particularly in the design, testing and documentation stages of the project. This strategy of evaluating the scope of impact of a change request also reduces project risk by restricting projects to a manageable size and focusing change on a restricted set of related business functions.

Software Asset Valuation

Function Point Analysis is being used increasingly by organisations to support the 'valuation of their software assets'. In the past, software has been considered an expense rather than a capital asset and as such was not included in an organisations asset register. The most commonly used software valuation method is based on the 'deprival method'. This method values the software based on what it would cost to replace in today's technical environment rather than what it cost originally to build. The industry build rate (dollar cost per function point) is determined and the total replacement value is calculated based on the current functional size of the application.

Since FPA provides a means of reliably measuring software then some organisations have implemented accrual budgeting and accounting in their business units. Under this directive, all assets must be valued based on deprival value and brought to account, thus ensuring better accountability of the organisations financial spending. Funding via budget allocation is based on assets listed in their financial accounts and their depreciation. In the past, the purchase price of the software recorded as an expense within an accounting year. These more recent accounting practices mean that it can now be valued as an asset and depreciated.

Publicly listed organisations have found that by using this accrual accounting method of measuring software as an asset rather than an expense they can amortise the depreciation over five years rather than artificially decrease the current year's profit by the total cost of the software. This strategy has a dramatic effect on their share price since once their software is listed as a capital asset it contributes to the overall worth of the company and the total cost of that asset has a reduced impact on the current year's reported profit.

Outsourcing Software Production and Support

The benefits of Functional size measurement in outsourcing contracts, is that functional size enables suppliers to measure the cost of a unit of output from the IT process to the business and enables them to negotiate on agreed outcomes with their client.

Specifically these output based metrics based on function point analysis has enabled **suppliers** to:

- quantitatively and objectively differentiate themselves from their competitors
- quantify extent of annual improvement and achievement of contractual targets
- negotiate price variations with clients based on an agreed metric
- measure financial performance of the contract based on unit cost of output
- at contract renewal be in a stronger bargaining position supported by an eststablished set of metrics

Conversely these output based metrics based on function point analysis has enabled **clients** to:

- Objectively assess supplier performance based on performance outputs delivered rather than concentrating on inputs consumed.
- Establish quantitative performance targets and implement supplier penalties and bonuses based on achievement of these targets
- measure the difference between internal IT costs compared to the cost of outsourcing based on similar output
- quantitatively compare competing suppliers at contract tender evaluation stage.

Most of the international outsourcing companies use function point based metrics as part of their client service level agreements. Whilst this method of contract management is relatively new its proponents are strong supporters of the usefulness of the technique. In our experience once an outsourcing contract has been based on Function Point metrics subsequent contract renewals expand on their use.

Metrics initiatives have a high cost and need substantial investment, which is often overlooked at contract price negotiation. Both the supplier and the client typically incur costs. However, given the size of the penalties and bonuses associated with these contracts it soon becomes obvious that this investment is necessary.

About the Author

Ms Pam Morris (B.Sc., Dip. Ed., Grad. Dip. Computing, CFPS), is the CEO for TOTAL METRICS Pty. Ltd Australia. She has extensive experience in the software development field, specialising in software process improvement and software metrics since 1989. She has consulted and presented to a wide range of organisations both in Australia, Japan, United States of America, New Zealand and the United Kingdom. Ms Morris is a founding member of the Australian Software Metrics Association (ASMA), holding a position on the Executive Board and the Function Point Counting and Benchmarking Database Special Interest Groups. Ms Morris is the international project editor of the ISO Standard 14143 for Functional Size Measurement and is convenor of WG12 (the ISO/IEC standards group responsible for the development of functional size measurement standards). She plays an active role internationally in the development of the FPA technique and has represented ASMA on the International Function Point User Group (IFPUG) Counting Practices Committee since 1993. She is a core member of the Common Software Measurement International Consortium (COSMIC).

… end …

Come Back Function Point Analysis (Modernised) – All Is Forgiven!

*Attributed by Charles Symons (Joint Leader with Alain Abran) of COSMIC, the **CO**mmon **S**oftware **M**easurement **I**nternational **C**onsortium and of Software Measurement Services (UK)*

ABSTRACT

Function Point Analysis was invented by Allan Albrecht of IBM as a means of sizing business application software independently of the technology used for its development. Albrecht's method was heavily promoted in its early years and has become the most widely used such method. However, the underlying model which Albrecht used for his sizing method, which was valid in the mid 70's when it was first conceived, is increasingly difficult to apply to modern software development. This and other factors have led to a decline in the method's use.

In this paper, we examine the reasons for the decline, and the main advances in thinking on the more general topic of Functional Size Measurement (FSM) which have been made in the last 15 years. Specifically, the COSMIC FFP method is seen as a significant step forward in being the first method designed by an international group of software metrics experts to work for both business application and real-time software.

Furthermore, it has been realised that a reliable FSM Method would be a very great asset with many uses, such as helping improve requirements specification, estimating, project 'scope creep', supplier performance measurement and contract control, etc. The experience of the new methods and the realisation of their potential value indicate that a return to popularity of (modernised) Function Point Analysis, in the guise of more general FSM Methods such as COSMIC FFP is highly likely.

1. BACKGROUND

The original idea of measuring a size of software from its requirements or functional specifications was introduced by Allan Albrecht of IBM over 20 years ago (Ref. 1). At the time it was a genuine breakthrough in thinking by providing the first method for sizing software which was independent of the technology to be used for its development. The method could therefore be used for comparing performance across projects using different technologies (using measures such as 'productivity', defined as size / effort) and as a first step in methods for estimating effort early in a project's life-cycle. This was a big step forward compared with the use of counts of Source Lines of Code (SLOC), which had been the only size measure up to that point.

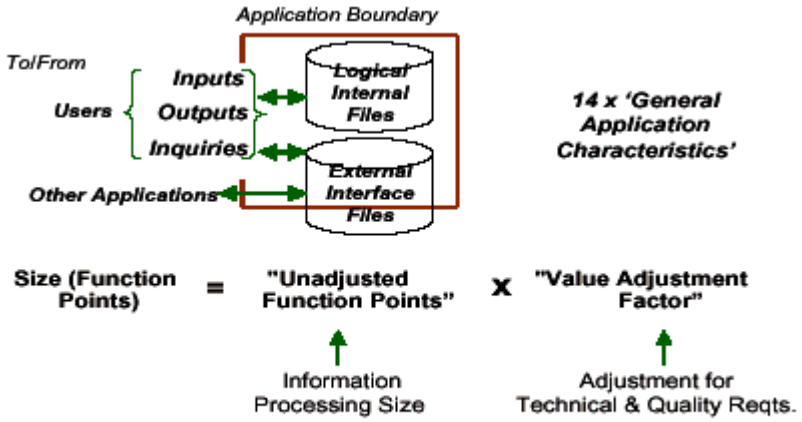

Fig. 1 The Albrecht (IFPUG) 'Function Point' model

Albrecht's model of functional specifications requires the identification of five types of components, namely input, output and inquiry elementary processes, and logical internal and external interface files. Having identified these five types of components in a specification, they are then weighted for complexity and are allocated 'unadjusted function points'. The total of 'UFP's for all components is then multiplied by a Value Adjustment Factor which takes into account the supposed contribution to size of some 14 technical and quality requirements (see Fig. 1).

Albrecht's Function Point method coincided with a push by IBM to promote 'programmer productivity' in its primary markets of business application, or 'management information systems' (MIS) software. The sizing model fitted well with how MIS software functionality was thought about in the mid 80's and the resulting performance measures were plausible. The method was heavily marketed by IBM's sales staff and software engineers, and a whole new micro-specialism of 'function point analysis' was spawned. Estimating tools were developed based on FPA and sold successfully, and within a few years an International Function Point User Group ('IFPUG') was established to take over responsibility for standards and promotion of the method.

The model of the five concepts and the Value Adjustment Factor is a pragmatic model. Albrecht was quite rightly trying to find a sizing method for MIS software which would correlate with development effort and it is a remarkable tribute to his ideas that the model has been successful for so long.

It seems safe to assert that in the late 80's and early 90's FPA was tried and used to some extent by the Information System departments of most major companies and Government departments in North America, much of Western Europe and other parts of the world. It is equally safe to say that the method is now much less widely used. It has in fact followed the well-known pattern of the Gartner 'hype-curve' (see Fig. 2) and only now is FPA climbing out of the 'trough of disillusionment' in new guises and with new realism.

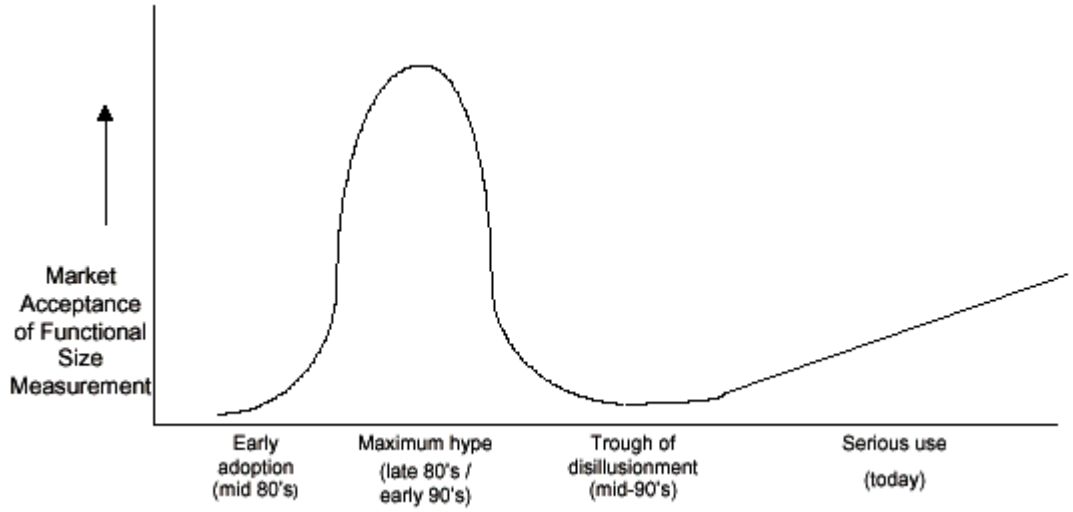

Fig. 2 The Gartner 'Hype Curve' for Functional Size Measurement

The duration of the trough of disillusionment has been long and harmful in the sense that the topics of performance measurement and early life-cycle estimating are quite frankly backwaters in the whole spectrum of software engineering methods. Recent graduates in computer science have heard of FPA, but very few ever find the opportunity to practice it in their real jobs. Progress in improving Functional Size Measurement or FSM (to give the generic name for the topic) following Albrecht's original lateral thinking has been extraordinarily slow. Understanding of performance measurement in software engineering is limited and every week we read reports of the consequences of poor early life-cycle estimating.

This paper examines the factors which led to the period of disillusionment, why it lasted so long, and the changes which are now taking place that are leading new 'modernised' developments of Albrecht's original FPA method into mainstream software engineering.

2. THE TROUGH OF DISILLUSIONMENT WITH FPA

Undoubtedly the reasons for the prolonged trough of disillusionment are much wider than just the limitations of Albrecht's original method. When the method was first promoted it was hyped as the software manager's latest 'silver bullet' which led to setting up comprehensive software metrics programmes in many organisations. These programmes turned out, however, to be more difficult to establish and to produce useful results than was anticipated. With the great benefit of hindsight, one can see the problems which arose.

- Often the gathering of software metrics data was delegated too low down in the organisation. Measuring the performance of software development projects, the results of which would reflect on the performance of hard-nosed project managers, is unavoidably a political matter. Junior metrics staff inevitably have a tough time getting the attention needed to collect reliable data. And if it's 'garbage in', then it surely follows there will be 'garbage out' and the programme will soon be discredited

- If the programmes did not fail due to poor input data, many failed due to poor analysis and presentation of the collected data. FSM methods are based on assumptions about reasonable approximations of size, and the results often tend to show a wide divergence of performance about the mean. Moreover, FSM is inherently a rather academic subject which is not easy to explain simply to a senior manager who may not have a background in software engineering. Such an IS Manager would receive comprehensive performance data from the Computer Centre showing how the processors were loaded and the data could be used to optimise performance and plan for upgrades. So why couldn't the software development group

produce convincing, meaningful data about their performance? The effect of poor analysis and presentation of a complex subject leads to suspicion about the validity of the measurements and averages. If senior IS management did not understand the results and/or could not see how to use them, the metrics programme would be an obvious candidate for the next round of budget cuts

- Albrecht's method has been very helpful for sizing software which is to be built to bespoke requirements, which was the dominant activity of Information Systems departments in the 70's and the 80's. But over the most recent decade many major companies switched over to using packaged software rather than creating their own bespoke software. Add in the adoption of rapid application or 'time-boxed' development, where estimating is partly forgotten about, and FP-sizing suddenly seems to be much less relevant.

- The continuous waves of mergers and acquisitions and outsourcing deals automatically led to cost-cutting and to loss of the FPA and software metrics expertise in many organisations. Ten years ago, for example UK Central Government departments all had significant software metrics expertise. Now that expertise has largely been outsourced.

- One might have imagined that entering into a contract to outsource software development and maintenance would automatically lead to an emphasis on measuring the performance of the supplier. But such contracts are negotiated by accountants and lawyers under great time pressure and often in secrecy from the organisation being outsourced. Whatever other skills they bring to the party, it is very rare to find an accountant or a lawyer who has any real understanding about measuring performance in software development. They tend to look for global cost reductions, open-book accounting, ways of agreeing costs in advance of any new project and such-like control methods.

All these factors have tended to diminish interest in FPA, but in addition the concepts of the IFPUG FP method itself now appear simply out-of-date when compared against the concepts of modern software development methods.

The IFPUG organisation has, of course, made valiant efforts to produce guidance on how to interpret Albrecht's original five component-types in terms of modern development methods. But inevitably if the developer is working in terms of concepts such as Objects or GUI's or Use Cases, or API's or whatever, the translation from these concepts back to the twenty-year-old concepts of 'Elementary Inputs', or 'Internal Logical Files', etc will seem like a distraction to his main purpose. And the Value Adjustment Factor is now totally irrelevant to modern software development.

One of the difficulties of having to continuously add new rules to cope with interpreting an old sizing method in terms of new development methods and technologies is that the method definition and procedures become increasingly complex. It becomes hard to maintain consistency of interpretation for all circumstances (see examples below). Once the size measurement method is no longer in the main stream of the project manager's or the developer's mind-set and methodology, and it becomes difficult to use, it will very easily be forgotten about. It is not at all surprising that Albrecht's successful pragmatic approach of 20 years ago is now running into difficulties.

3. THE EMERGENCE OF 'SERIOUS USE' OF FUNCTIONAL SIZE MEASUREMENT (FSM)

After the hype has died down, we now see clear signs of growing and serious use of FSM, albeit updated in new forms. For this to happen, several conditions must be satisfied.

- There must be clear potential value in the market-place for reliable, practical FSM Methods

- The FSM Methods must be compatible with modern ways of developing and maintaining software and must be demonstrably usable for sizing and estimating. There must be a clear understanding and confidence in what these methods can and cannot do

- If organisations which have built up significant repositories of performance data and estimating methods based on a particular sizing method are to be persuaded to adopt a new sizing method, there must either be conversion rules which will help migration from the old to the new sizing method, or the need to change must be so compelling that the organisation is willing to make a significant investment in the new sizing method

- There must be a truly international quorum of expertise in the subject to provide training, consultancy, user support, tools, standards, etc., that is, all the paraphernalia which a user of these methods will need before he is prepared to invest in them. The software industry is global; local methods are of little interest to major corporations

- The problems cited above of setting up, sustaining and gaining benefits from software metrics programmes must be addressed

- Finally, there must be sustained marketing to make the market aware of the potential value and of the availability of proven solutions

We will examine each of these factors in turn.

4. THE MARKET POTENTIAL FOR RELIABLE FSM METHODS

Supposing we have a reliable, practical method of producing a size of software early in its life based on functional requirements, what might be the potential uses? The 'mind-map' (Fig. 3) shows the potential of using functional size measurement as a component of methods –

- for the requirements analyst to produce requirements that are measurable (as well as unambiguous, structured, traceable, testable, etc)
- for the project leader to assist with estimating, control of requirements 'scope creep', etc
- for investors to help decide on the cost/benefit analysis for new projects and for valuing software assets at their replacement value
- for those concerned with performance improvement to measure and benchmark performance
- for procurement and contract managers to help select a supplier and to measure and control its performance.

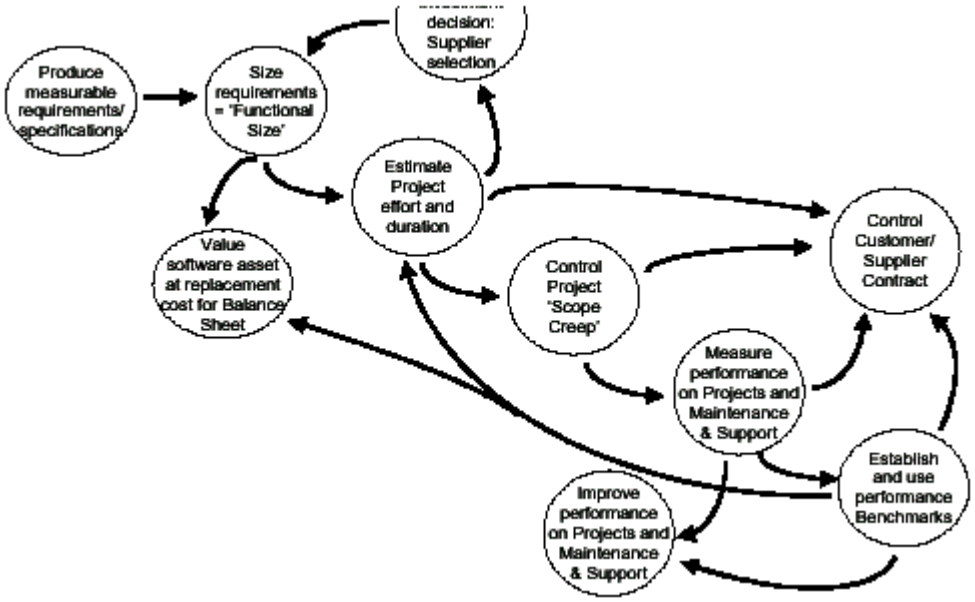

Fig. 3 'Mind-map of the uses of Functional Size Measurement

There is clear market potential demand for all of these potential uses. There is no need to remind readers of this paper of the need to improve early life-cycle estimating, of the needs of project leaders to control requirements, and so on. Even those involved in outsourcing software development and maintenance, having gone through the transition processes of the early contract years, are now increasingly concerned to improve methods of controlling value for money from their suppliers. Some Government Departments, for example, (Ref. 2) are specifying the use of FSM Methods as part of their software procurement procedures.

All these remarks apply to the world of business application or MIS software, for which Function Point Analysis was originally designed. The world of 'real-time' software (by which we mean software for telecoms, process control, operating systems, embedded systems, avionics, etc) has never experienced generally available FSM Methods, so the potential value in that segment of the software industry is even greater.

The economic value of the potential for reliable FSM Methods is therefore undeniably enormous.

5. 'MODERNISING' FUNCTIONAL SIZE MEASUREMENT

Fig. 4 below shows the development of some of the principal ideas for improving Functional Size Measurement since Albrecht's original FPA method.

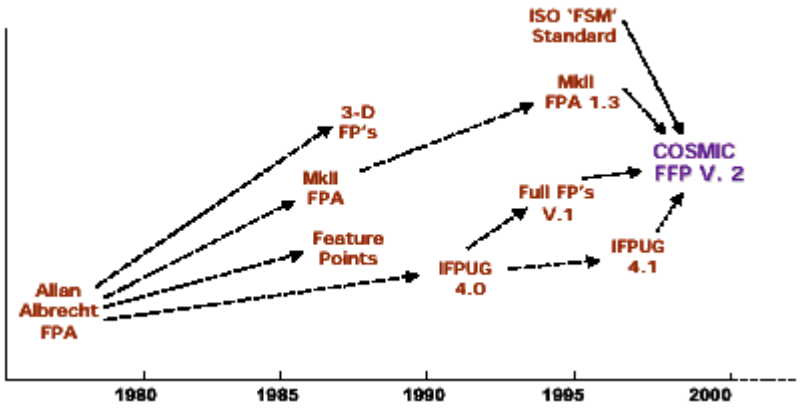

Fig. 4 Evolution of Functional Size Measurement

Albrecht's original FPA method has evolved over the last 20 years into a method now known as 'IFPUG 4.1', though the original basic concepts and weighting methods have not changed since 1984. Over this same period other methods have been put forward, each attempting to overcome weaknesses perceived in Albrecht's original model, or to extend its field of application. Noteworthy amongst these are the following.

- Jones (Ref. 3) extended Albrecht's model in his 'Feature Points' method by adding a sixth component to account for the contribution to size of mathematical algorithms. This method has not really caught on due to the inherent difficulty of agreeing standard ways of defining and assigning a weight to algorithms of increasing size and complexity. It seems fair to say that this problem of sizing mathematical algorithms in a simple way is a problem which has not yet been solved

- MkII FPA (Ref. 4) was proposed by the present author to improve the functional size scale (by comparison with the Albrecht method) so that it takes better account of the internal processing complexity of MIS software. The method also updated some of Albrecht's basic concepts in line with 'Structured Analysis' methods, which were popular in the late 80's and early 90's. The MkII method regards MIS software requirements as composed of 'logical transactions' each with an

input, process and output component. The size of the 'process' component is measured by counting the number of entity-types referenced in the processing of the logical transaction. One of the main differences between the Albrecht and MkII methods therefore is that the former method counts 'Logical Files' once per piece of software being measured, whereas the latter method counts 'entity-types' every time they are referenced in each logical transaction. The MkII method originally proposed an extended Value Adjustment Factor, but this was dropped a few years ago when it was realised it was no longer meaningful.

- 3-D Function Points were proposed by Whitmire of Boeing (Ref. 5) as a means of extending Albrecht's method into real-time software. Whitmire added 'Control' components to Albrecht's 'Functional' and 'Data' components. It is understood that the 3-D Function Point method is still used successfully in Boeing, but details of the method have not been published outside Boeing.

- Full Function Points, or 'FFP V1' was developed by a team based at the University of Quebec in Montreal, Canada (Ref. 6). This method relies on the IFPUG method to size MIS software, but adds six new component-types to enable sizing of real-time software. The method has been extensively and successfully trialled for sizing real-time software by several industrial partners. Its development has now ceased in favour of the COSMIC FFP method.

- In 1996 the International Standards Organisation started a Working Group (ISO/IEC JTC1 SC7 WG12) on Functional Size Measurement, with the participation of software metrics experts from about ten nations. The Working Group has concentrated on establishing the common principles of functional size measurement, rather than attempting to develop a new FSM Method. The fist publication setting out these basic principles was ISO/IEC 14143-1, published in 1998 (Ref. 7).

- At the end of 1998, a group of software metrics experts, mostly participating in the ISO Working Group established COSMIC, the Common Software Measurement International Consortium. This group set out to develop a new FSM Method which would work equally well for MIS and for real-time software. The principles of the new 'COSMIC FFP V2' method were published in October 1999 (Ref. 8), field trials were held in 2000, and the results of these trials have recently been published (Ref. 9). The COSMIC FFP model for functional sizing is extremely simple, see Fig. 5. In outline, Functional User Requirements are decomposed into 'Functional Processes' which in turn can be decomposed into 'FunctionalSub-Processes'. A Functional Sub-Process is a type of Data Movement, namely an Entry, an Exit, a Read or a Write, each assumed to have associated data manipulation. The method explicitly does not claim to measure the size of functionality which includes complex data manipulation (i.e. algorithms), and does not attempt to take into account the effect on size of Technical or Quality requirements.

What has been learned and what progress has been made over this last 20 years of endeavour? In the author's opinion, the following are the most important lessons learned. (In the following, the detail of much of the supporting argumentation has had to be omitted through lack of space.).

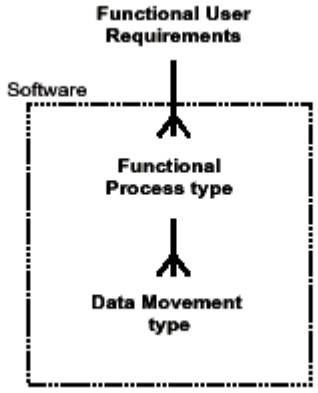

Fig. 5 The COSMIC FFP functional sizing model

5.1 The need for Unambiguity

First, no functional sizing method will ever be successful unless it is based upon completelyunambiguously defined concepts. Unless this condition is satisfied, different analysts will producedifferent sizes from the same requirements document or functional specification. Most importantly, theMkII Logical Transaction and the COSMIC FFP Functional Process share the same conceptual basis (asequence of sub-processes, triggered by a unique event-type outside the software, which, whencomplete leaves the software in a coherent state with respect to the event). The IFPUG definition of anElementary Process is now close to this same concept, which originates with the work of the foundersof 'Structured' methods in the 60's and 70's.

As an illustration of the ambiguity associated with some development methods, Jensen of IBM GlobalServices reported that he had discovered a very large number of variations of interpretation of the UML'Use Case' concept within IBM. After much analysis, Jensen and the present author concluded that theMkII Logical Transaction (or COSMIC FFP Functional Process) is a specific case of a Use Case.Exploiting this idea as a refinement of using Use Cases leads to establishing statements of requirementswith a far higher chance of unique interpretation, and which are measurable (Ref. 10). The MkIILogical Transaction is now a basic component of the IBM LEAD estimating method for Object-Orienteddevelopments (Ref. 11).

One could continue with other examples. The IFPUG 'Internal Logical File' concept is frequently oneof the most difficult concepts to interpret in practice. The COSMIC FFP Data Movement, in contrast,is tightly defined; difficulties of ambiguous interpretation were not experienced in the field trials.

5.2 The need for Domain Independence

Second, it is vital to have a functional sizing method which works equally well in the MIS and real-time software domains. Although every experienced software engineer has an instinctive understanding of the difference between these two main broad categories of software, it is actually very difficult (as the ISO Working Group has discovered) to make a clear distinction. And in real-world software engineering, the two domains overlap.

MIS systems frequently receive real-time data feeds or have to interface with real-time software such as for telecoms. And telecoms and process control software may well need to access significant databases which have all the characteristics of MIS software. Similarly, real software projects in both domains frequently involve developing or changing software items in different layers or peer items of multi-tiered infrastructure software architectures. Practitioners concerned with sizing and estimating need common methods which will work across both these two main domains, in any layer or peer software item.

The COSMIC FFP V2 method is the first publicly-available such method which meets these criteria.

5.3 The need for Different Sizes for Different Purposes

Third, we need different sizes for different purposes, depending on the scope and purpose of the sizing activity. When Albrecht invented the Function Point method his aim was to define a size measure of business application software seen from the viewpoint of the human user's requirements. He also wanted a method which could produce a size with reasonable analysis effort, so inevitably the sizing model is a simplified view of the full detail of how a human user interacts with the software.

For many years this human view of the business application coincided with the application developer's view. So 'first generation' IFPUG and MkII Function Point sizes have been accepted as valid measures of a development team's 'work-output' and therefore likely to correlate with the required effort to develop the software for a given technology

Even when PC / main-frame client-server architecture was first introduced, the rules for FPA could be adapted to cope with the new views. Metrics specialists recognised the need to make a distinction between the human user (or 'end-user') view of the size, and the developer's view. The end-user's view does not change simply because the functionality is distributed over two processors (the end-user is unaware of the distribution) so FPA methods could still be applied to measure size from this view. But they could also be applied to measure the size of the client and the server components separately (see Fig. 6 below). And metrics specialists could explain that the sum of the sizes of the two components measured separately would exceed the size as seen from the end-user viewpoint, due to the inter-processor communication.

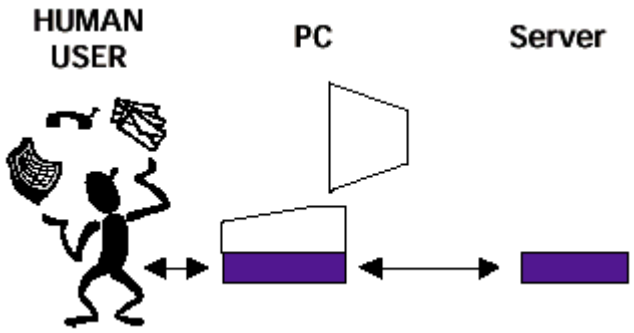

Fig. 6 Two-tier (PC/server) Architecture

For performance measurement purposes, metrics specialists could still use the end-user view of size, and it is interesting to note that performance on projects measured this way often appears lower than if the same functionality were developed on a single platform. But if we are concerned with accuracy, particularly of estimating, it is desirable to use the sizes of the two components measured separately, especially when the two components are developed using different technologies. For this level of architectural complexity, first-generation methods such as IFPUG and MkII FPA are still perfectly adequate.

But these methods have really reached their limit of applicability when it comes to measuring and estimating for a multi-tier, multi-layer software architecture.

The main reason why a major bank joined the COSMIC FFP field trials is shown in Fig. 7 below. This shows the architecture of changes that had to be made to several of their systems to allow everyone in the bank to access a common customer database. For this purpose, the bank had to build enquiries at the PC's in the branches and offices, upgrade the back-office servers to handle the enquiries, upgrade a front-end processor which handles all traffic over the Wide Area Network, and build new, two-layer software on the mainframe to manage the common customer database.

When this system was sized with the IFPUG method (and the same would be true of the MkII method), the bank could measure only the functionality as seen by the human end-user at the PC, or perhaps at most the functionality of the PC component and the database server. The large amounts of functionality which the project team had to develop to support the end-user requirements are simply invisible and cannot be measured by first-generation FPA methods, without stretching the methods beyond what they were designed for. The IFPUG measurement therefore correctly sized the functionality as seen by the human end-user, but grossly under-sized the functionality delivered by the software team. The measurement appeared to show that the team had a very low productivity.

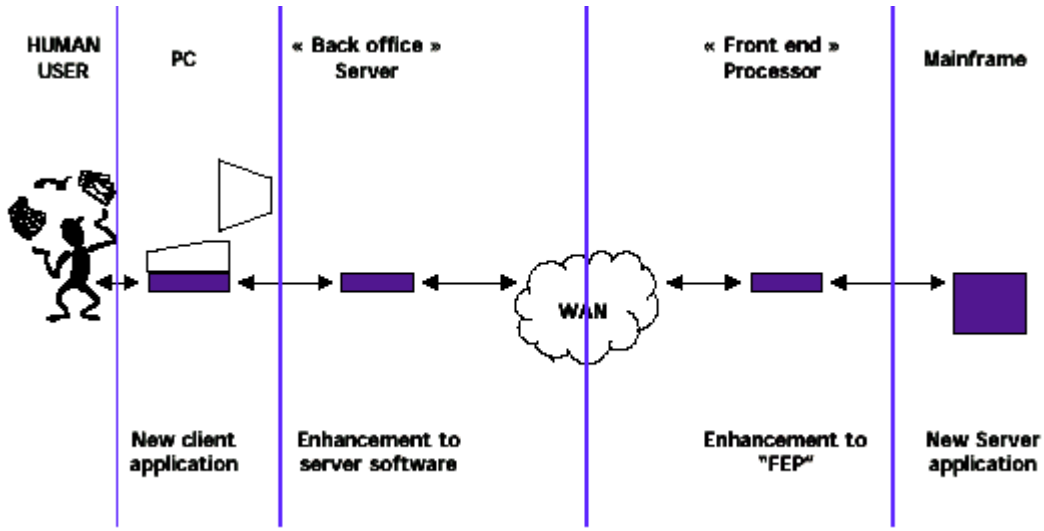

Fig. 7 A four-tier client-server architecture

With the 'second-generation' COSMIC FFP method, all five components of the system could be separately sized, which of course gave a much larger and more realistic view of the functionality delivered from the developers' viewpoint.

Generally the power of the COSMIC FFP method is that it can measure the functional size of any component of software as seen by the *direct user(s) of that component.* And a component can be at any level of decomposition. In COSMIC FFP terms, a 'user' is 'any person, engineered device, software component or thing which interacts with the software being measured'. So in Fig. 7, the user of the back office server is the PC in the branch, and so on.

For future purposes of producing standard size measurements, we may need to distinguish

- The 'external user' view of functional size – a size measure mostly of interest to the human user, only in the MIS domain
- The size of the 'principal components', that is the size of the principal separately developed and maintained sub-systems, maybe also running on separate processors
- The size of any 'building brick' components which have to be built.

The reason to distinguish these 'size types' carefully is that they are not additive. Just as the size of a set of bricks cannot be added to obtain the size of a wall, so the size of a set of objects cannot be added together to obtain the size of a principal component assembled from those objects.

5.4 The Way to deal with Technical and Quality Requirements

The fourth lesson is that Albrecht's approach to account for Technical and Quality requirements via a Value Adjustment Factor which multiplies the size obtained from pure Functional User Requirements is not the best way to deal with such requirements.

Albrecht's approach is rather like saying we will estimate the size of a house in cubic meters, then adjust this size using a scale running from 'Not Present' to 'Significant Impact', for a series of factors such as whether the building site has easy access or not, whether materials are scarce, whether the kitchen needs fitting out, etc. These factors clearly need to be taken into account when estimating the *effort* and *cost* to build the house, but they do not alter its *size*.

Similarly, Functional Size should be based purely on Functional User Requirements. Technical and Quality Requirements should be taken into account in methods to estimate development effort, cost and elapsed time, since their impact clearly depends on the technology to be used (see Ref. 11 for a

good example of how Technical and Quality requirements are handled).

Summarising the main lessons learned from developments in FSM Methods over the last 15 years.

- The concepts on which modern FSM Methods are defined, such as the COSMIC FFP method must be, and are believed to be, unambiguously defined, so as to ensure consistent sizing
- FSM Methods must be 'domain independent'; this condition is satisfied at least for MIS and real-time software by the COSMIC FFP method, but no FSM Method is currently capable of sizing complex mathematical algorithms
- We need different functional sizes for different purposes. The human user view of size is quite different from the size seen by the developer of specific components of a system. The same FSM Method may be used at the various levels of abstraction, but the resultant sizes may not be additive
- Technical and Quality Requirements should not be combined with Functional User Requirements to produce a composite size. The result is very difficult to interpret, and the scale factors inevitably depend on assumptions about the technology to be used

6. CONVERSION FROM OLD TO NEW FUNCTIONAL SIZE SCALES

Where two functional size scales attempt to measure the same size, one may reasonably expect a simple conversion formula between the two scales. Such is the case for conversion between the IFPUG Unadjusted FP scale and the MkII FP scale, now that the latter method has discarded a Value Adjustment Factor. Measurements of the 'external user' view of size of the same software carried out using both methods have produced a simple quadratic conversion formula (Ref 12).

Similar experiments using the FFP V1 method showed a close relationship of size with the IFPUG scale, when they were both measuring the functionality as seen by the external user.

But clearly, if one method can only measure the external user view of size whilst another method can measure the size of components of the same software, then no conversion is possible from the former measurements to the latter. (The reverse may be possible, but this needs more research.)

Organisations with large accumulations of measurements of software using the IFPUG or MkII methods who wish to be able to measure the size of components of their infrastructure using a method such as COSMIC FFP V2 therefore have to accept that there can be no conversion from the old measurements to the new. However, it may still be perfectly reasonable to continue to use the 'old' measurement methods for certain purposes, and to add in the 'new' measurements for the new purposes.

7. INTERNATIONAL SUPPORT FOR THE COSMIC FFP METHOD

A specific goal of the COSMIC initiative has been to 'develop, test, bring to market and gain acceptance' of the new FSM Method. This goal is being pursued in several ways.

First the COSMIC Core Team now comprises software metrics experts from Australia, Canada, Japan, Vietnam and six European nations. Field trials of the method were conducted on three continents. The Measurement Manual has been translated from English to French and Spanish, whilst German, Italian and Japanese versions are in an advanced stage. The COSMIC FFP method has been accepted onto the work programme of ISO and an 'ISO-compatible' version of the Measurement Manual will be tabled in May 2001.

Training course and case study material has been developed and training is now available in various parts of the world. The performance measurements from the field trials (mostly from real-time software projects) are being made available to the International Software Benchmarking Standards Group. The Measurement Manual, much case study material and research papers are freely available in the public domain from the University of Quebec at Montreal web-site (www.lrgl.uqam.ca).

In summary, therefore, although the COSMIC project still has a long way to go, a great deal of effort is being deployed towards this goal and much has been achieved over the last two years.

8. IMPROVEMENTS TO AND MARKETING OF SOFTWARE METRICS PROGRAMMES

This goal will also take some time to achieve, but there are significant forces driving towards improving understanding of the processes needed to produce reliable software performance metrics and to use those data for estimating, performance improvement and all the other potential uses identified above. Amongst the most important drivers are the following.

- Efforts in the direction of Software Process Improvement, particularly via using the Capability Maturity Model (or CMM). The new CMM-I model requires attention to measurement at Maturity Level 2, rather than ML 4, as in the original CMM model. (Ref. 13)
- The new ISO 9001 (2000) places greater emphasis on measurement than the previous version
- As stated above, there are increasing signs of awareness amongst clients of outsourced software development and maintenance services of the need for accurate measures and benchmarks of their supplier's performance. Similarly, as more software is contracted out and margins are squeezed, suppliers become more aware of the need for early life-cycle estimating
- National and International Software Metrics Associations and commercial conference organisers continue their good work of holding conferences which promote software process improvement and software metrics. National SMA's in particular are doing good work in developing standards for software metrics where none existed before, and in improving functional size measurement (for examples, see Refs. 14 and 15)
- Providers of estimating tools, software metrics repositories, and project management tools (e.g. for task planning and effort recording) are delivering more and better integrated solutions
- The COSMIC Core Team is seeking and taking every opportunity to promote its new method and to open up its use

All these developments auger well for more attention and more professionalism being given to software metrics in general and to Functional Size Measurement in particular.

9. CONCLUSIONS

We have traced the development of Functional Size Measurement from a very clever and pragmatic beginning through its inevitable hype, followed by disillusionment, and now its development and maturing in new forms. This maturing, particularly with the COSMIC FFP method, has the potential to make significant contributions to software requirements engineering, project management and estimating, software process improvement, software procurement and contract control, and to improving the understanding of the economics of software as an asset.

In this process of maturing, considerable new insights have been gained into software size measurement. The need now is for software managers and engineers to build on this new knowledge and experience, to tackle the many challenges that still lay ahead, to experiment with and promote these methods, and to reap the benefits for the software industry.

Come back Functional Size Measurement, all is indeed forgiven!

Software Measurement Services Ltd
143 High Street
Edenbridge, Kent TN8 5AX
England.Page 12
Tel: +44 1737 763 674

Fax: +44 1737 761 166
E-mail: charles_symons@compuserve.com
Web: www.software-measurement.com

REFERENCES

1. A.J. Albrecht, "Measuring Application Development Productivity', presented at IBM Applications Development Symposium, Monterey, CA, 1979
2. A.L. Rollo, T Wright, 'Southern SCOPE – a Method for Acquiring Custom-Built Software, or Let the Customer Manage Software Acquisition', ESCOM Conference, London, April 2001
3. C. Jones, 'A Short History of Function Points and Feature Points', Software Productivity Research Inc., USA, 1987
4. C.R. Symons, 'Function Point Analysis: Difficulties and Improvements', IEEE Transactions on Software Engineering, 14(1), pp 2-11, 1998
5. S.A. Whitmire, '3D Function Points: Scientific and Real-time Extensions to Function Points', Pacific Northwest Software Quality Conference, 1992
6. A. Abran, J.M. Desharnais, M. Maya, D. St-Pierre, P. Bourque, 'Design of a Functional Size Measurement for Real-Time Software', Research report no. 13, Software Engineering Management Research Laboratory, Universite du Quebec a Montreal, Canada, November 1998
7. ISO/IEC 14143-1:1998. 'Software Measurement – Functional Size Measurement – Part 1 Definition of Concepts'
8. A. Abran, J.M. Desharnais, S. Oligny, D. St-Pierre, C.R. Symons, 'COSMIC FFP Measurement Manual, Version 2.0, Ed. S Oligny, , Software Engineering Management Research Laboratory, Universite du Quebec a Montreal, Canada, October 1999. Downloadable at www.lrgl.uqam.ca/ffp.html
9. A. Abran, C.R. Symons, S. Oligny, 'An Overview of the COSMIC FFP Field Trial Results', ESCOM Conference, London, 2001
10. P.G. Rule, 'Using Measures to Understand Requirements', ESCOM Conference, London, 2001
11. G. Jensen, 'Estimating Life-Cycle Effort and Duration', Australian Conference of Software Measurement, November 2000
12. C.R. Symons, 'Conversion Between IFPUG and MkII Function Points', obtainable from www.software-measurement.com
13. 'Capability Maturity Model – Integrated. Continuous representation ver 0.2b, IPD additions, Software Engineering Institute, Carnegie Mellon University, USA
14. UK Software Metrics Association ('UKSMA'), 'Quality Standards: Defect Measurement Manual, Release 0.i', obtainable from www.uksma.co.uk
15. Netherlands Software Metrics Association ('NESMA'), 'Definitions and Counting Guidelines for the Application of Function Point Analysis, Version 2.0, 1997

… end…

News On Measurement Standards

ISO Standard for Functional Size Measurement

From: ISO Working Group 12

What is it? - Background to Functional Size Measurement

Organisations engaged in software engineering have struggled for years in search of acceptable quantitative methods for measuring process efficiency and effectiveness, and for managing software costs, for the systems they acquire, develop, enhance or maintain. One critical, and particularly elusive, aspect of this measurement requirement has been the need to determine software size. Numerous software sizing methods have been proposed in the past. These included numbers of source lines of program code and various measures derived from the technical characteristics of the software.

These methods have limitations in that they can not be :

- applied early in the software development process,
- applied uniformly throughout the software's life time,
- easily interpreted in business terms, or
- meaningfully understood by users of the software.

The concept of *Functional Size Measurement* (FSM) overcomes these limitations by shifting the focus away from measuring how the software is implemented to measuring size in terms of the functions required by the user. The first method to develop this concept was Function Point Analysis developed in 1979 by A. Albrecht of IBM.

Since the public release of Function Point Analysis, many sizing methods have been developed based on its concepts. Differing interpretations of the original concepts has resulted in inconsistencies amongst the various sizing methods. These inconsistencies lessen the ability and attractiveness of any of these methods to be a used as a standard method for the functional sizing of software.

Background to the FSM Standard

Late 1992 the software metrics User Groups from Australia, UK, Netherlands and the USA created a workgroup under ISO/IEC/JTC1/SC7.. A suite of standards was proposed as part of the WG12 Project 7.31 to resolve these inconsistencies and establish a more rigorous method for functional sizing. As the standards have progressed the following user groups have actively contributed to their development.

- Australian Software Metrics Association (ASMA),
- Centre d'Intérêt sur les Métriques (CIM),
- Deutschsprachige Anwendergruppe für Software Metrik und Aufwandschätzung (DASMA),
- French Function Point Users Group (FFPUG),
- Gruppo Utenti Function Points Italia (GUFPI),
- International Function Point Users Group (IFPUG),
- Netherlands Software Metrics Association (NESMA),
- UK Function Point Users Group (UFPUG).

Status of the Standards

ISO/IEC/ JTC 1/ SC 7/ Working Group 12 (WG12) administers the 7.31 project for Software Measurement: Functional Size Measurement. The Project is made up of five component parts:

- Definition of Concepts
- Compliance Assessment of Software sizing methods to ISO/IEC 14143-1:1998
- Verification of a Functional Size Measurement Method
- Functional Size Measurement Reference Model
- Determination of Functional Domains for use with Functional Size Measurement.

Part 1 of this standard ISO/IEC 14143-1:1998 was published June 1998 as a full International Standard is available from the National Standards Bodies within each country. Part 2 of this standard is currently being balloted as a Committee Draft (CD) version of an International Standard Part 3 , 4 and 5 are being balloted as Preliminary Technical Reports (PDTRs). Technical Reports are not as rigorous nor prescriptive as full International Standards since they usually represent an area which is still evolving and has not yet settled sufficiently to be 'set in rock' as a standard. It is anticipated that these Technical Report will evolve into full International Standards over the next five years.

Structure of FSM Standards

The following sections describe each of these standards and how they assist users of Functional Size Measurements achieve the stated benefits.

Part 1: Definition of Concepts
Part 1 identifies the common fundamental characteristics of functional size measurement methods and defines a set of generic mandatory requirements for a

method to be called a Functional Size Measurement Method (FSM). The role of this part of the standard is to promote the consistent interpretation of FSM principles and facilitate comparability between Functional Size measures. It is anticipated that the main users of this part 1 will be those persons who are involved in developing functional size measurement methods and need to know their fundamental characteristics. People who need to verify that a particular software sizing method complies with the fundamental concepts of an FSM may also use it.

Part 2: Compliance Assessment of Software sizing methods to ISO/IEC 14143-1:1998

Part 2 establishes a framework for assessing the extent of compliance of a particular functional size measurement method with the mandatory requirements of Part 1 of this standard. It aims to ensure that he outputs from this compliance assessment process are objective, impartial, consistent, repeatable and correctly represent he characteristics of the functional size measurement method being assessed.
Through the introduction of this standard, benefits anticipated are:

- Compliance assessors will have specific procedures to follow when assessing the extent of compliance of a software sizing method with Part 1.
- Provision of a framework for all the inputs, processes and outputs of a compliance assessment for a functional size measurement method againstPart1.
- Enable the production of a standardised output report from a compliance assessment which will allow users of a software sizing method to evaluate the strengths and weaknesses of different methods and select he one which is most suited to their needs.
- Assistance to developers of potential Functional Size Measurement Methods in complying with the mandatory requirements of Part1.
- Increase User awareness of the relative merits of FSM Methods.

Persons needing to verify that a particular software sizing method complies with the fundamental concepts of an FSM will use part 2 of this standard.

Pam Morris (Australia) is the project editor. Peter Fagg from the UK is the co-editor.

Part 3: Verification of a Functional Size Measurement Method

Although a number of functional sizing methods are currently in use worldwide there is no framework with which their effectiveness as measurement techniques can be expressed. This standard provides the process and the criteria against which the functional size can be validated. It is aimed as assisting designers and promoters of functional size measures to meet the specified criteria and to state their level of conformance to measurement theory and practice.
Through the introduction of this Technical Report benefits anticipated are that it will:

- Provide a framework procedure for use when users or developers need to verify the capability of an FSM Method.
- Users of a FSM will be able to evaluate the strengths and weaknesses of different FSM Methods and select the one most suited to their needs.
- FSM Methods will have a tool, which will assist them in refining their technique to be more effective as a sizing method.

- Reduce the frequency of inappropriate claims of some current sizing methods
- Increase end user awareness of the relative merits of FSM Methods as measurement techniques.

Part 3 of will be used by persons needing to check that a particular FSM Method is a effective measurement technique. Canada's representative, Professor Alain Abran is the project editor for Part 3.

Part 4: Functional Size Measurement Reference Model

Users of the various functional sizing methods have made various claims regarding their limitations or usefulness when applied to different types of software such as management information systems, real time and scientific software. However there is no current agreement on Standard Reference sets of Functional User Requirements against which such claims can be verified. The purpose of this project is to provide reference points against which users of the validation process can assess the effectiveness of an FSM for different software types in various software environments. Through the introduction of this Technical Report benefits anticipated are:

- Developers of an FSM Method will be able to test the functional domains for which their method can be used effectively and refine their method
- Persons verifying a Functional Size Measurement Method will be provided with reference objects against which an FSM can be applied and compared.
- Reduction in the inappropriate use of some current sizing methods
- Better comparison of data used in quality and productivity benchmarking.

Persons needing to verify that a particular FSM Method is an effective measurement technique and persons needing example sets of functional user requirements from different functional domains will use part 4 of this standard.

The project editor for this part of the standard is Professor Eberhard Rudolph of Bremerhaven University in Germany. The assistant editor is Mitsuhiro Takahashi of Japan.

Part 5: Determination of Functional Domains for use with Functional Size Measurement

Users of the various functional sizing methods have made various claims regarding their limitations or usefulness when applied to different types of software such as management information systems, real time and scientific software. However there is no current agreement on the characteristics of the Functional User Requirements which would classify them into different functional domains. The different terms describing software types have only been loosely defined by convention. The purpose of this part 5 is establish a standard for classifying functional user requirements for use in the application of FSM.
Through the introduction of this Technical Report, benefits anticipated are:

- Users of FSM will be able to evaluate the characteristics of their functional user requirements and categorise them as belonging to one or more functional domains
- Select the FSM Method, which has been shown to be appropriate for those functional domains relevant to the Functional User Requirements to be sized.

- Developers of an FSM Method will be able to clearly state the functional domains for which their method can be used effectively.

- Compliance testers and assessors of functional size measurement methods will be provided with clear guidelines for determining the functional domains for which their method is effective.

Persons needing to check the effectiveness of their FSM Method or check the compliance of functional size measurement methods will use part 5 of this standard. It may also be used to assess the functional domain of a particular set of functional user requirements so as to determine which FSM Method is most appropriate for their needs.

The project editor for Part 5 is Carol Dekkers of the USA. Mary Bradley who is the current chair of the IFPUG Counting Practices Committee is the co-editor.

It is anticipated that work will continue on these standards over the next two to three years. If you would like more information about the standards please contact the WG12 Convenor (Pam Morris) at Pam.Morris@Totalmetrics.com or contact the project editors listed below.

Contacts

WG12 Convenor Pam Morris Email:mailto:Pam.Morris@Totalmetrics.com
Project Editor - Project 7.31, Sub-project 7.31.2
Pam Morris Email:mailto:Pam.Morris@Totalmetrics.com
Project Editor - Sub-project 7.31.3
Jean-Philippe Jaquet Email : jacquet@fermat.ciepcct.uenf.br
Project Editor - Sub-project 7.31.4
Eberhard Rudolph Email : erudolph@HS-Bremerhaven.DE
Project Editor - Sub-project 7.31.5
Carol Dekkers Email : dekkers@compuserve.com

Industry Events

AsiaSTAR 2002 Conference

Call for Papers and Presentations
AsiaSTAR 2002 Conference

Software Testing Analysis and Review

July 22-24, 2002
Melbourne, Australia

Theme: Building on Experience

A message from the program chairs:

Software Testing is a challenging profession and today's testing professionals are faced with

- a growing numbers of technologies and lifecycles,
- increasingly complex integrated solutions,
- an extensive range of tools and techniques,
- difficult (market-driven) time constraints, and
- economic rationalisation.

Yet, we are expected to remain effective in our testing methods and efficient in our process of doing so.

AsiaSTAR 2002 offers an excellent opportunity to meet with other testing professionals to share

- proven practical ideas and experiences (through case studies),
- the latest theories and innovative solutions, and
- success stories or heroic defeats.

We therefore invite you to be an integral part of this event by submitting a technical paper.

The theme for this year's conference is ***Building on Experience***. Therefore, we plan to bring together a multitude of experienced testing practitioners and industry leaders to discuss not only leading-edge techniques, but also real-world approaches to meeting today's testing challenges.

We invite you to share with us your knowledge and experience of effective software testing techniques, methods and technologies that have benefited your organisation. We also invite you to present new ideas and insights into how to turn theory into practice.

Donna O'Neill; Colin Cherry..Program Chairs

AsiaSTAR 2002 Call for Papers and Presentations

Proposal submission deadline 1 February 2002

We are offering speakers the opportunity to actively participate in the second AsiaSTAR conference. AsiaSTAR 2002 will draw upon the experience of our highly successful conference last year to create an exciting and informative event that is even bigger and better.

We invite you to submit a presentation proposal for the AsiaSTAR 2002 conference based on your experiences and insight into software testing issues. Proposals should reflect the conference theme of ***Building on Experience***, and may cover any aspect of software testing and evaluation including:

Technology:

Internet and associated technologies
PC-based applications
Mainframe and Client Server solutions
Object Oriented solutions
Embedded systems
Real-time systems
Safety-critical systems

Techniques & Processes:

Test Process Improvement
Test Design Techniques
Requirements Management
Test Automation
Usability Testing
Performance, Stress & Load Testing
Risk-based testing
"Good enough" techniques
Improving testability
Technical reviews and Inspections
eX-treme (agile) testing

Management & Organisation:

Influencing organisations and management
Test Planning and Controls
Risk Management
Measuring the success of testing
Defect Tracking and Reporting
Test Team Organisation
Test Team Staffing Issues
Training and Development

Vendors please note:

Our policy is to avoid product-specific talks in the main technical programme (ie, it is not intended as a forum for presenting the features of specific products or services). However, vendors are invited to submit presentations for the main programme as long as the talk reflects a technical topic rather than promotes a particular product. There will be separate Vendor-only time slots available for product-specific talks.

Submission Information

The standard time slot for presentations is 45 minutes, including 10 minutes for questions. Double-length presentations (90 minutes) will also be considered.

One complimentary conference registration is provided for each accepted presentation.

The submission process is as follows:

1. Prepare a summary of your presentation. The summary should be 1 to 2 pages long and should identify at least *3 key points* that you intend to cover. The summary should clearly indicate the topic's relevance to your audience and to the conference theme of ***Building on Experience***. Choose a title that accurately describes the contents of the presentation. Note that if you already have a completed presentation, submit the presentation rather than a summary.
2. Complete the attached speaker application form.
3. Submit the speaker application form and presentation summary by **1 February 2002** to the AsiaSTAR 2002 headquarters. Submissions may be made by email to kate@einsteinandedison.com.au, by fax to +61 2 6232 4245, or via the AsiaSTAR conference web site at http://www.testingconferences.com.
4. You will be notified of selection by **22 March 2002**.
5. If your submission is successful, you will need to submit the final paper, presentation slides, etc by **10 May 2002**. We encourage speakers to prepare technical papers as well as a slide presentation, to provide conference attendees with additional details, supporting material and references.

Notes on Submitting a Presentation

- When submitting a presentation for consideration, it is important that you provide all requested information. The contact information you provide with your submission will be used for all future correspondence and should reflect a location where you can be reached throughout the conference planning period. Please notify us if your contact information changes.

- If required by your organisation, get approval for your presentation before submitting it. Also, verify that funding is available for you to attend the conference.

AsiaSTAR 2002 Speaker Application Form

Submit this page and presentation / summary to the AsiaSTAR 2002 program chair by **1 February 2002**:

- Email to kate@einsteinandedison.com.au; or
- Fax to +61 2 6232 4245.

1. Speaker Information

Name _____

Company _____

Position _____

Email _____

Phone _____ **Fax** _____

Address _____

_____

State _____ **Post Code** _____

Country _____

Previous experience as a speaker (if any) _____

_____

_____

2. Presentation Information

Title _____

Duration (45 or 90 minutes)

Key Points 1. _____

2. _____

3. _____

3. Intended Audience

Position (Management, Practitioner etc.)

Level (Entry, Intermediate, Advanced)

ACOSM 2001

Some of the topics covered in ACOSM2001, were:

- Measurement and Analysis in Software Process Improvement: A Practical Guide to
- Applicable Standards and Models.
- Things out of Control? 10 easy metrics that give you value today!
- What to do when the experts disagree
- A Survivors Guide to Maintaining a Metrics Program
- Validation of Application Results of COSMIC-FFP to Switching Systems
- Software Project Estimation – Black Art of Science
- We don't need process …. Do we?
- Changing the Relationship between Business and Developers: SouthernSCOPE
- Getting more 'bang for your buck' from your Function Point Counters

For copies of Conference material of ACOSM 2001, held in November, contact Robyn Smith on:

asmavic@ozonline.com.au

About ASMA

ASMA NATIONAL - for membership enquiries

Administration Support: Robyn Smith

Postal Address ASMA
PO Box 127
Warrandyte VIC 3113

Phone: 03 9844 0560 Fax: 03 9844 0561 Mobile: 0411 649909

Email: **asmavic@ozonline.com.au**

WEB PAGE ADDRESS OF ASMA, ISBSG AND IFPUG:-

HTTP://WWW.ASMA.ORG.AU
HTTP://WWW.ISBSG.ORG.AU
HTTP://WWW.BANNISTER.COM/IFPUG/HOME/DOCS/IFPUGHOME.HTML

STATE CHAPTERS - ALPHABETICALLY

ASMA ACT

Chairman-Oliver Hague

Postal address: ASMA ACT
PO Box 287
Civic Square. ACT. 2608

Phone: 06 259 1540 Fax: 06 244 7963

Email: **cipher@atrax.net.au**

ASMA NSW

Administrative Support: Maria Patane
Postal address: ASMA NSW
PO Box 432
Leichhardt. NSW. 2040

Phone: 02 9716 7525 Fax: 02 9716 7525 Mobile : 0416 205070

Email: **asma@cia.com.au**

ASMA NSW 2001 Executive Committee:

POSITION	NAME	EMAIL	WORK(02)
Chairman	Rocco Muller	**rm@softwarebestpractice.com**	0411 464 464
Treasurer	Maria Patane	**asma@cia.com.au**	9716 7525
Committee Member	Gerard Neill	**gneal@westpac.com.au**	
Committee Member	Ian Brightwell	**ianbri@dh4.com.au**	
Committee Member	Larry Langman	**llangman@woolworths.com.au**	

ASMA QLD

Chairman-Bruce Hodgen

Postal address: ASMA QLD
PO Box 5220
Alexandra Hills. QLD. 4161

Phone: 07 3875 3768 Fax: 07 3823 3169

Email**:** **B.Hodgen@cit.gu.edu.au**

ASMA VIC

Administration support - Robyn Smith

Postal Address ASMA
PO Box 127
Warrandyte VIC 3113

Phone: 03 9844 0560 Fax: 03 9844 0561 Mobile: 0411 649909
Email: **asmavic@ozonline.com.au**

ASMA VIC 2001 Executive Committee

POSITION	NAME	EMAIL	WORK
Chairman	Paul Radford	**radfordp@charismatek.com**	03 9696 1255

Program Director	Michael Stringer	**m.stringer@sagecomp.com.au**	03 9888 1760
FSM SIG	Pam Morris	**pam.morris@totalmetrics.com**	03 9882 7611
ISBSG SIG	Terry Wright	**terry.wright@mmv.vic.gov.au**	03 9651 9006
Committee	Paula Holmberg	**pholmberg@vmcsnat1.telstra.com.au**	03 9963 2122
Committee	Robyn Lawrie	**lawrier@charismatek.com**	03 9696 1255
Committee	John Olgivie	**mmaogil@au.ibm.com**	03 9963 2137
SEA Metrics	Kate Hartwig	**MissKate@bigpond.com**	03 9686 3232

.. ……end of Metrics Matters……