

asma
Innovative IT thinking

Metrics Matters

The Australian Journal of
Software Metrics

May 2002

CONTENTS

ABOUT METRICS MATTERS	3
SOFTWARE METRICS ARTICLES	4
The Trials & Tribulations of Function Points.....	4
The Power of COSMIC FFP.....	8
ISBSG – Collecting Valuable Data	11
Uses and Benefits of Function Points: Part 4.....	13
New Approaches to Software Metrics	16
NEWS ON MEASUREMENT STANDARDS	26
Who could use ISO 9000?	26
RESEARCH CORNER	27
Evaluating Software Measurement.....	27
INDUSTRY EVENTS	29
ACOSM 2002	29
IFPUG CFPC Test.....	30
2002 Software Conferences.....	31
METRICS RELATED INFORMATION SOURCES	32
Reference Books.....	32
Interesting web sites	33
ABOUT ASMA	34

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.

Permission is granted to reproduce any item from Metrics Matters on the condition that Metrics Matters and ASMA are fully acknowledged as the source.

Software Metrics Articles

The Trials & Tribulations of Function Points

By
David Wright
DST International

Presented at ASMA Victoria Metrics Forum Meeting on 27 March 2002.

In January this year we finally made function point counting a regular and required task in our development methodology. All designs are now function point counted.

This has been the culmination of a journey that began over 2 years ago.

It has been a journey that has met with much resistance – and in quarters I did not expect and ways I did not expect. It has been a journey that has demanded a lot of focus, persistence and energy. But it is a journey I believe we had to make.

DST International is a software company that specialises in marketing and developing software to/for the Funds Management industry. Some of you will have heard of us – our flagship product is the HiPortfolio Investment Management System.

The prime development centre for HiPortfolio is the Melbourne Development Centre. We (DSTi) also have development shops in Bangkok, Boston and Surbiton. Surbiton and Melbourne are the major development centres.

In year 2000 we decided to implement function points in to the Melbourne Development Centre. There were (and still are) two noble aims. **Firstly we wanted function points to provide an independent measure for productivity.** Secondly, we believed the counting process would provide a quality assurance check point

Until function points we measured our productivity on a concept of 'billable days'. This was a measure of how many approved project days we could get through. The measure had worked well. Our people became more focussed on the tasks on hand and our time utilisation increased enormously. It drove major improvements in our productivity – and quality. If you were working on bugs you couldn't get billable days – so don't create bugs!

But, at the end of the day, billable days was a SUBJECTIVE measure. And it did not have 'in built' motivations beyond quoting accuracy and time management. It was a cost plus measure.

We were looking for something more; something that provided:

- objectivity in measurement
- motivation to improve the software process
- a measure of the benefit delivered to the client as opposed to just the effort

Function points seemed to fit the bill.

The second objective was quality assurance. We are constantly working on improving our design statements. The simple theory was that if we couldn't count the points on the design, the statement wasn't up to scratch.

So we thought the objectives were clear; they were honourable objectives. We even told people about these objectives and why. We even heard people say this sounded better than billable days!

Our journey was planned as a four-step implementation approach. To begin, train everyone in the concepts of function points; and for a selected few train them in counting the points. Secondly, count the baselines for each of our products. Thirdly, perform a year's analysis so that we had a bench mark year and finally, implement the function points for performance measurement. It all seemed pretty logical and straightforward.

Training is where we hit the first snags. We had professional trainers come in and do a days presentation. Comments and concerns soon started to surface.

'Function Points can't be measured accurately'

'How can we measure performance when the effort between individual function points varies so much?'

'Our work is complex, there is no way we can use function points for estimating'

I then compounded the doubts by making a big mistake. I suggested that we may be able to use function points as the basis of personal performance.

The quiet resistance to function points turned into open rebellion!

But after much discussion and persuasion, we got through that. The next stage was counting the baselines. We did this work in conjunction with a metrics specialist company – ‘Total Metrics’. The people assigned to the team to count the points were keen and enthusiastic and the project got off to a flying start. But as the project progressed these keen and enthusiastic people came to a major conclusion - They didn’t want to count points!

They certainly saw the project through but it had a profound impact

- one guy left the department
- another left just several months later

And they had made everyone else gun shy. It had been agreed, though, that part of the duties of a business analyst was to count points – so we had more conscripts sent to the function point front!

I, by this stage was starting to experience small bouts of depression and overwhelming desires to strangle people.

The purpose of the Historical Analysis phase was to count function points on all design statements produced for 12 months. This would give a baseline year of metrics to which subsequent years could be compared.

This did seem a straightforward task until the Gremlins arrived.

- ‘Joe’s away sick – can’t count the points’
- ‘We have urgent requests to get other design statements done – will pick up the counts after that’
- ‘The backlog is too big’
- ‘Our people aren’t trained’
- ‘This counting is just slowing us down’
- ‘The baseline is out of date’
- ‘It’ll take months to catch up the backlog why don’t we re-baseline and start again next year?’

It was a war of attrition and I was getting the strong impression the guys weren’t with me on this one!

When all seemed lost the Testing Manager suggested that we let his team count the points. He had people who were keen to learn the skills. In 6 weeks his team cleared that huge backlog.

But we had to face an old issue again. The Testing Manager could not keep the same people counting points. Even though these people had been keen and enthusiastic, their interest inevitably waned.

Our solution has been to outsource the counting. Since then we have been counting the points on each design and are well in to the final phase – performance measurement.

A development shop can be measured on its ‘performance triangle’ – quality, time and cost. You want to produce good software quickly and cheaply. We can now measure and monitor our performance. In fact our bonus system is now based on achieving those objectives. Our targets are measured in support cost per function point (good quality), development cost per function point (cheap cost) and average delivery time per function point (quick time).

This ability to target our performance and then measure ourselves against this performance will create some powerful forces for change.

Since introducing billable days we have trebled our billable days output with the same number of people. The number of bugs reported back from clients has halved and the total number of bugs waiting to be fixed at any time is a 1/6th of what it was 5 years ago.

What has caused this? Stricter controls? Better procedures? The fear factor? No! Targets and regular feedback.

Billable days have taken us this far – but it can go no further.

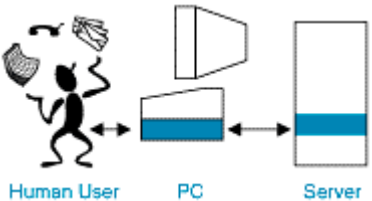
Function points are our next wave. Using these targets and feedback, I have no doubt that the guys will find ways to beat the targets, slash the costs, reduce the times and improve the quality. With the right information they will have the capacity to amaze! This is what the function point journey has been all about - the reason to overcome the skepticism and obstructions; the complaints and the difficulties.

Metrics, based on function points, will be the key that unlocks our next great wave forward.

The Power of COSMIC FFP

Reproduced with permission from Software Measurement Services, from Issue 8 of their 'Software Measurement & Process Newsletter' – see www.software-measurement.com.

The Power of COSMIC of software as seen by decomposition. In COSMIC, the functional size of any component can be at any level of any person, engineered device, other software component or the component being measured.



Recent work has demonstrated its power to measure the size of any component of multi-tier, multi-layer software, at any level of decomposition. This opens up new opportunities for:

- improving the accuracy of measuring performance
- early life cycle estimating
- measuring software development from any domain (business, real-time, infrastructure, etc).

In the beginning

When Allan Albrecht invented Function Point Analysis he aimed to measure business application software size, as viewed from the requirements of the human user. He also wanted a method that was cheap and easy to use. Inevitably the sizing model is a simplified view of how a human user interacts with the software.

For many years the user's view coincided with the developer's view. So IFPUG and MKII Function Point sizes have been accepted as valid measures of a development team's work output. Hence their successful use for many years in performance measurement and estimating.

Even when PC / main-frame client-server architecture was introduced, FPA adapted to cope with the new views. Metrics specialists recognised the need to make a distinction between the human, or end, user's 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. 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. 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.

For performance measurement purposes (ie the effort and time to develop software), we could still use the end-user view of size. However, performance on projects measured this way often appears lower than if the same functionality were developed for a single computer. For accuracy, particularly of estimating, it is desirable to use the sizes of the two components measured separately, especially when they are developed using different technologies.

The difficulties of measuring and estimating for two-tier client-server systems are nothing compared with those for multi-tier, multi-layer software architectures.

Now there is COSMIC

A major UK bank joined the COSMIC field trials because they had problems sizing systems like the one shown below. They wanted to allow all their staff to access a common customer database. The bank therefore had to build enquiries at the PCs in the branches, upgrade back-office servers to handle the enquiries, upgrade the front-end processor (handling traffic over the WAN) and build new, two-layer software on the mainframe to manage the common customer database.

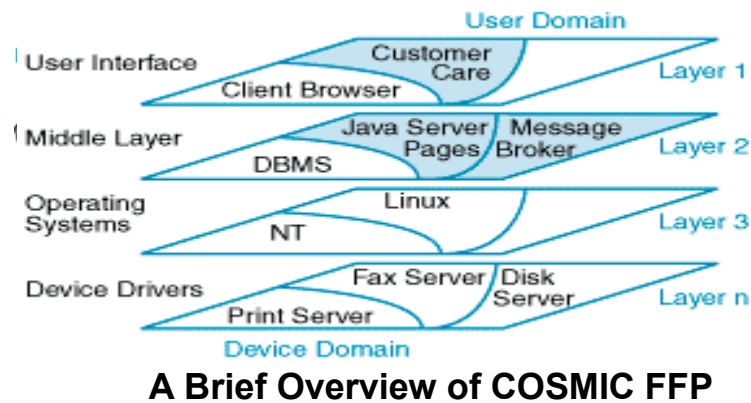
When this system was sized with the IFPUG method (the same would be true of the MkII method), it measured the functionality as seen only by the human at the PC. The large amounts of functionality which had to be developed to support the end-user requirements were invisible and cannot be measured by conventional function point methods. The IFPUG method therefore grossly under-sized the functionality delivered by the software team, resulting in an apparently very low measurement of their productivity.

With the COSMIC FFP method, we were able to size all five components of the system which, of course, gave a much larger and more realistic view of the functionality delivered.

A unified theory

COSMIC is based on solid theory and decades of international experience. It has been designed from the outset both to comply with the ISO standard for Functional Size Measurement (ISO 14143) and to be compatible with modern ways of specifying requirements (e.g. use cases and prototyping). It is applicable to software developed using software engineering principles such as abstraction, encapsulation and information hiding, and works for projects using COTS (Commercial Off The Shelf) software and component integration.

The COSMIC Field Trials have demonstrated that COSMIC measures the parts that other methods cannot reach.



11

Figure 1. A generic model of software

The method recognises that modern software development makes use of components at different levels within a software architecture.

To size the total functionality to be delivered, identify the requirements of each component that will be created or modified. Identify the functional processes involved for each, then identify and count the data movement types, which are entries and exits across the boundary to the user (human, machine or software), and the reads and writes to storage.

11

Figure 2. Layered view of software architecture

Fig 2 shows the development of a customer care web front end. As can be seen different requirements have been allocated to software at different levels. Enhancements are required to the existing Customer Care System and Message Broker components. Java Server Pages also need to be developed.

ISBSG – Collecting Valuable Data

By Peter Hill, ISBSG

Every one wants information but no one wants to collect data! Obviously that statement is not strictly true, as the **International Software Benchmarking Standards Group**, (ISBSG), has managed to gather data on close to 1,400 software projects from more than 20 countries. But far more people want to devour information from the ISBSG analyses, than provide the data on which the analyses are based. One of the roles of the ISBSG is to provide a data collection “package”. This package is effectively the data collection “standard” that presents questions to gather software project data that can be stored in the ISBSG Repository and then used for analysis.

What can we discover from the data that is collected?

What is an optimum team size for maximum productivity?

Does productivity improve with the use of CASE tools?

What development techniques are being used and what has been their impact?

Is outsourced software development more successful than software developed internally?

What accuracy of software estimation are companies experiencing?

These are just a few of the many questions that software professionals are seeking answers to. The data collected by the ISBSG allows analysis to turn this data into information that provides the answers.

But the ISBSG data collection package is not restricted to the ISBSG itself. Any organisation can download a copy of the package from the ISBSG web site, (www.isbsg.org.au) and use it as the basis of their own project data collection. There is no charge for the download as the ISBSG wants to encourage as many people as possible to adopt its data collection “standard”. Naturally the hope is that this initiative will result in an even greater number of projects being sent to the ISBSG.

The ISBSG has just revised its collection package for New Developments & Enhancements. The new package will be available for download in April. A data collection package for Maintenance and Support will be released later in the year.

Both of these packages have been developed by an international working group, thereby ensuring broad acceptance of the data collection “standards”.

For more information on these packs please contact Robyn Smith on 03 9844 0560 or email her on asmavic@ozonline.com.au

Here is a list of the ISBSG Deliverables in 2002/03

The ISBSG plans to deliver the following over the next 12 months:

- | | |
|--|------------|
| 1. Organisational Benchmark Report | Jan 2002 |
| 2. Software Estimation Workshop – (PowerPoint workshop material) | Feb 2002 |
| 3. New ISBSG Web Site | April 2002 |
| 4. New Development IFPUG/NESMA new data collection package | April 2002 |
| 5. New Development MKII new data collection package | May 2002 |
| 6. New Development COSMIC new data collection package | May 2002 |
| 7. The Software Metrics Almanac (book in the style of Benchmark Release 5) | June 2002 |
| 8. Maintenance & Support – data collection package | Nov 2002 |
| 9. The Benchmark Release 7 (book in the style of Benchmark Release 6) | Nov 2002 |
| 10. Estimating Benchmarking & Research Suite (CD - Data Disk R8) | Jan 2003 |

... end ...

Uses and Benefits of Function Points: Part 4

Package Customisation

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

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).

INTRODUCTION

This is the last part of a four part series published over three issues of Metrics Matters. Each section has focussed 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 has looked 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**

4. Customising Packaged Software

4.1 Background

For selected MIS applications, implementing a packaged ‘off the shelf’ solution is the most cost effective and time efficient strategy to deliver necessary functionality to the business.

All of the benefits and uses of Function Point Analysis which applied to in-house development projects as described in the previous articles can also apply to projects which tailor a vendor supplied package to an organisations specific business needs.

Experience shows that Function Point Counting of packages is not always as straightforward as sizing software developed in-house, for the following reasons:

- only the physical and technical functions are visible to the counter. The logical user view is often masked by the physical implementation of the original logical user requirements.
- in most cases the functional requirements, functional specifications, and logical design documentation are not delivered with the software. The counter may have to rely on the User Manual or online help to assist in interpreting the user view.
⇒ *The modelling of the logical business transactions often requires the function point counter to work with the client to identify the logical transactions. They do this by investigating the users functional requirements and interpreting the logical transactions from the package's physical implementation.*
- in most cases the names of the logical files accessed by the application's transactions are not supplied by the package vendor.
⇒ *The function point counter will need to develop the data model by analysing the data items processed by the application.*

However, with sufficient care a reasonably accurate function point count of packaged applications can usually be obtained.

4.2 Estimating Package Implementations

The project estimates for a package solution need to be refined for each implementation depending on the percentage of the project functionality which is:

- native to the package and implemented without change
- functionality within the package which needs to be customised for this installation
- functionality contained with the organisations existing applications which needs to be converted to adapt to the constraints of the package
- to be built as new functions in addition to the package functions
- to be built to as new functions to enable interfacing to other in-house applications
- not to be delivered in this release.

The productivity rates for each of these different development activities (to implement, customise, enhance or build) are usually different. This complexity of assigning an appropriate productivity factor can be compounded when the package provides utilities which enable quick delivery based on changes to rule tables. Change requests, which can be implemented by changing values in rule-based tables, can be implemented very efficiently

compared to a similar user change request, that requires source code modification. It is recommended that these different types of activities are identified and effort collected against them accordingly so that productivity rates for the different activity types can be determined.

The functions can be flagged for their development activity type and their relative contributions to the functional size calculated. This will enable fine-tuning of the project estimates.

Another area of concern when developing estimates for package integration is the need to determine the extent that the application module needs to interface with existing functionality. The function point count measures the external files accessed by transactions within this application. A high percentage of interface files (>10%) suggests a high degree of coupling between this application and existing applications. A high degree of interfacing tends to have a significant negative impact on productivity rates and needs to be considered when developing estimates.

5. Summary


Function Point Analysis is a technique that until now has been restricted within many organisations to only be used for better estimating or input into benchmarking productivity rates. The above examples illustrate a wider range of uses where it can contribute to the better management and control of the whole software production environment.

... end ...

New Approaches to Software Metrics

Slides of a presentation by Ted Smilie of Montrose Computing
and Rocco Muller of Software Best Practise
at ASMA NSW Presentation on 4th March 2002 in Sydney.

Slide 3




Software Metrics

"We must stop messing with 'internal' software metrics such as complexity and function points and learn really powerful software metrics based on final-product, customer perceived results such as adaptability, availability, reliability, maintainability, security, portability and performance.

Tom Gilb

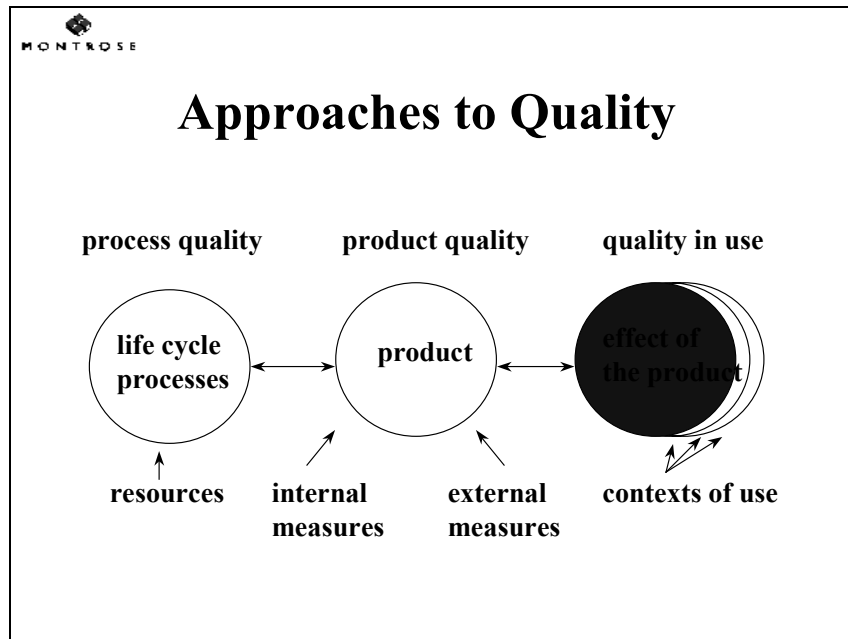
Slide 4



Product Vs Process Metrics

- **Traditional approach:** software product is too complex; best we can do is measure process - after all, product certification of a toaster only says it will not electrocute you, not that it will make good toast. Traditional models include CMM, ISO 9001, SPICE
- **Modern approach :** we can measure software product by identifying quality characteristics, sub-characteristics and metrics, during development, before handover and in use, e.g. ISO 9126 and ISO 14598

Slide 5




Slide 6

The slide, titled "International Software Engineering Standards" and featuring the MONTRÖSE logo, lists two international standards:

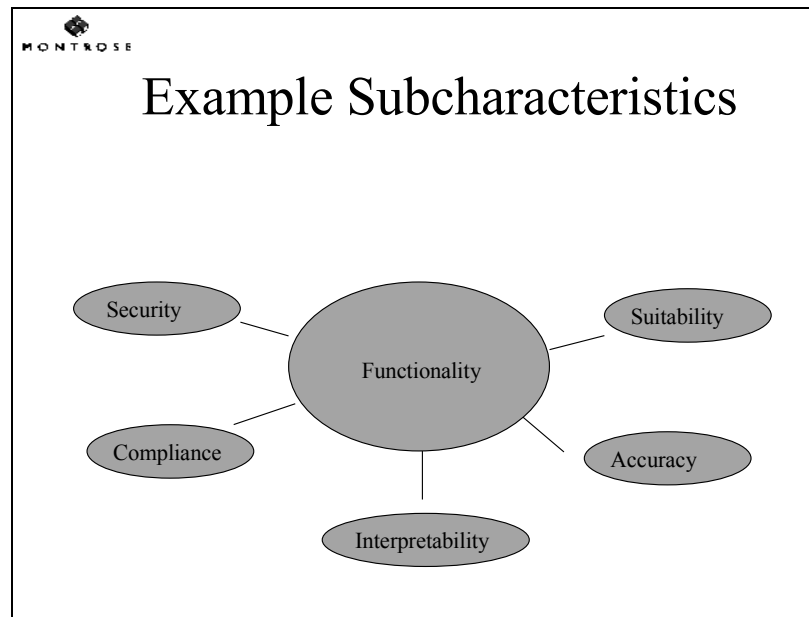
- ISO/IEC 9126, Information Technology - Software Quality Characteristics and Metrics
- ISO/IEC 14598 Information Technology - Software Product Evaluation

Slide 7


 **Software Quality Characteristics**

- Functionality
- Reliability
- Usability
- Efficiency
- Maintainability
- Portability

Slide 8




Slide 9

 MONTROSE

Sample External Metrics - Time Behaviour

- Response Time (average in period)
- Turnaround Time (average in period)
- Throughput (ratio of completed/un-completed tasks)

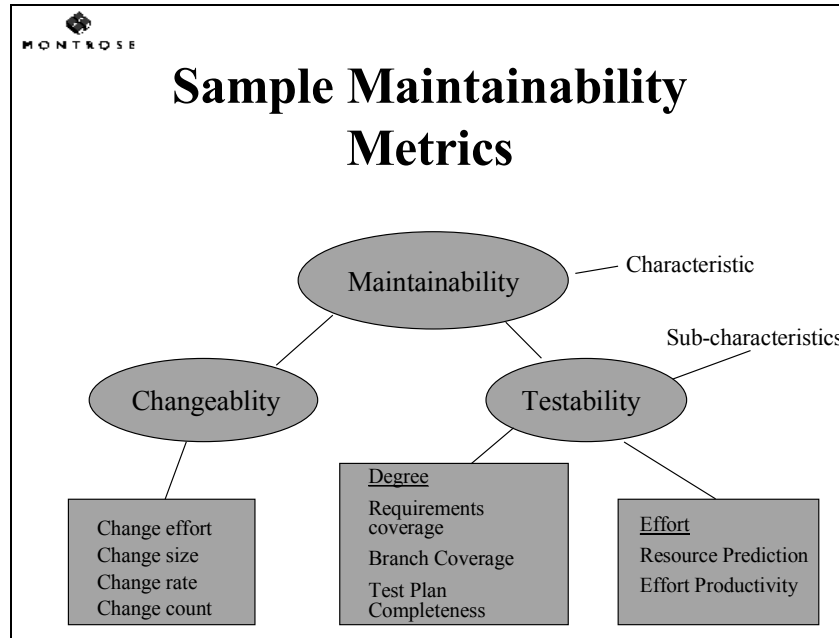
Slide 10

 MONTROSE

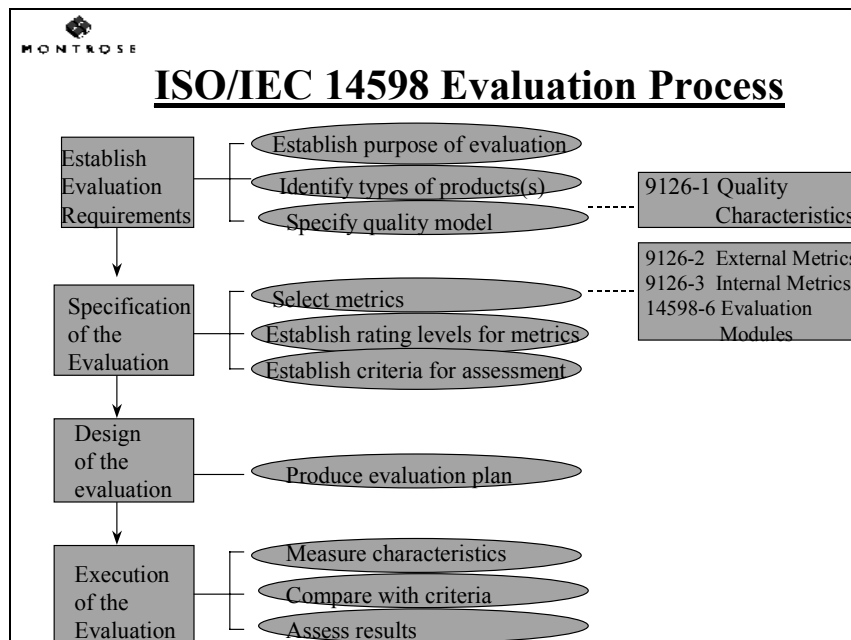
Sample Internal Metrics - Analysability

- Adherence to Programming Standards (ratio of modules with code review)
- Documentation (ratio of major functions documented)
- Diagnostics (ratio of modules with diagnostic functions)


Slide 11



Slide 12




Slide 13

 **Rated Level (ISO CD 14598-6)**


Level	Safety	Economy	Security	Environmental
A	many killed	financial disaster (out of business)	protection of strategic data & services	unrecoverable environmental damage
B	threat to human lives	large economic loss	protection of critical data & services	recoverable environmental damage
C	damage to property, few injured	significant economic loss	protection against error risk	local pollution
D	small damage to property, no risk to people	negligible economic loss	no specific risk identified	no environmental risk

Slide 14

 **SQuaRE**

- SQuaRE: Software (Product) Quality Requirement and Evaluation
- Re-structure of ISO 9126 and ISO 14598 Series.
- Concept - Lifecycle view, quality model, metrics & measurement, management


Slide 16

 **Five Metrics Categories.**

- early indications of problems,
- the quality of the products,
- the effectiveness of the processes,
- the conformance to the process, and
- the provision of a basis for future estimation of cost, quality, and schedule.

Software Project Managers Network


Slide 17

 **Example Project Metrics**

- Budget (actual against planned);
- Deliverables against planned milestones;
- Functionality delivered (on time and late);
- Risk exposure (internal, external and residual risk);
- Staff turnover (actual against planned and key staff retained); and
- Overtime (actual against planned).

www.e-envoy.gov.uk

Slide 18




Cost & Schedule Estimates

“More projects are doomed by poor cost and schedule estimates than by technical, political or team problems.”

William H. Roetzheim

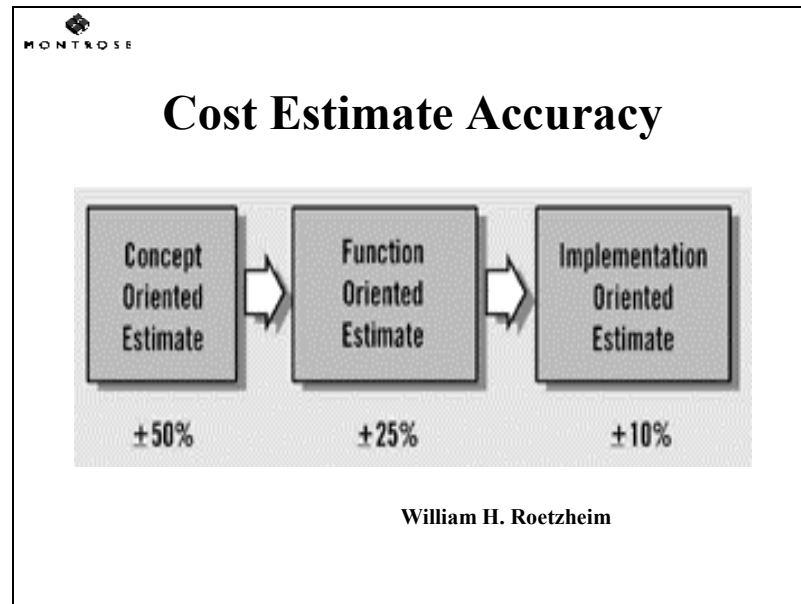
Slide 19



Estimation Techniques

- Function Point Counting
- Estimating size of source code - per thousand lines (KLOC)
 - Consensus of experts (Delphi), historical comparison
 - function points analysis and backfiring
- Constructive Cost Model (COCOMO), Barry Boehm


Slide 20



Slide 23

The slide, titled "Bibliography (1)", is enclosed in a rectangular frame. The logo for "MONTRÖSE" is in the top left corner. The title is centered at the top in a large, bold, black font. Below the title, there is a list of four references, each preceded by a bullet point. The references are: "Software Project Managers Network, http://www.spmn.com/", "Major IT Projects Review, Successful IT: Modernising Government in Action. http://www.e-envoy.gov.uk/publications/", "Estimating Software Costs, William Roetzheim www.sdmagazine.com/articles/2000/0010/0010d", and "The Agile Organization, Tom DeMarco,".

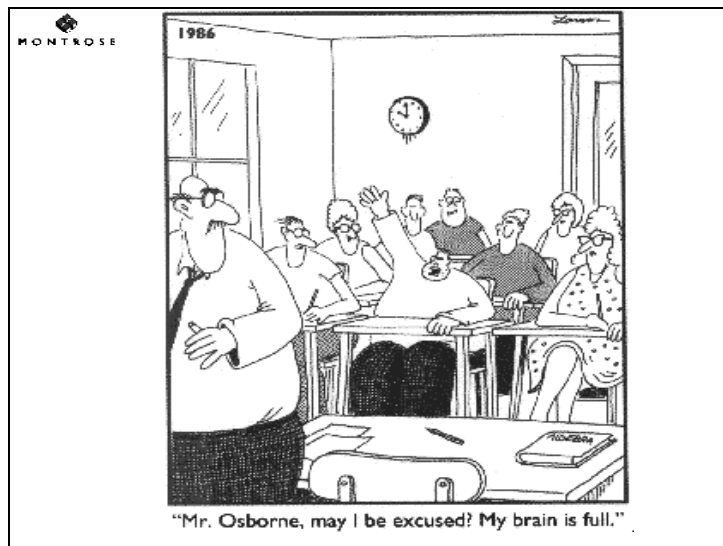
Slide 24



Bibliography (2)

- SquaRE The next generation of the ISO/IEC 9126 and 14598 , Motoei AZUMA
www.escom.co.uk/conference2001/papers/azuma.pdf

Slide 25



..... end

News On Measurement Standards

<h2>Who could use ISO 9000?</h2>

ISO 9000

This International Standard describes fundamentals of quality management systems, which form the subject of the ISO 9000 family, and defines related terms.

This International Standard is applicable to the following:

- a) organizations seeking advantage through the implementation of a quality management system;
- b) organizations seeking confidence from their suppliers that their product requirements will be satisfied;
- c) users of the products;
- d) those concerned with a mutual understanding of the terminology used in quality management (e.g. suppliers, customers, regulators);
- e) those internal or external to the organization who assess the quality management system or audit it for conformity with the requirements of ISO 9001 (e.g. auditors, regulators, certification/registration bodies);
- f) those internal or external to the organization who give advice or training on the quality management system appropriate to that organization;
- g) developers of related standards.

Research Corner

Evaluating Software Measurement

*By Mike Berry
University of New South Wales*

If measurement is important - the key to efficient and effective management and software processes, then why do so many organisations find it difficult to establish sustainable measurement? Why is measurement the first thing to be dropped when times are hard, instead of the last?

Some suggest that one reason is that many measurement frameworks fail to deliver value in terms of providing information to help people carry out the things they must do. The measurement frameworks may be churning out measures and analyses, but too often they are only “looking under the lamplight”.¹ So, how can an organisation ensure that its measurement framework is offering value?

Mendonca and his colleagues present one methodology for “understanding the data and the metrics and how they are fulfilling the needs of data users in an MF [Measurement Framework]”.^{2,3} Their approach addresses the twin issues of data being collected for no good purpose and of data being unused because its existence is unknown. Their premise is that “an MF should be sound, complete, lean and consistent”, where:

- Sound means the metrics and measurement models are valid
- Complete means the MF measures everything it users need to achieve their goals
- Lean means only what is needed is measured and nothing more
- Consistent means the metrics and scales are suitable for the user needs and can be applied when and where the user requires.

¹ A drunk comes out of a pub one dark night and drops his keys. He gets down on his knees and starts to feel for them. Not having much success, he moves over to the nearest lamppost and searches there. His mate comes along and asks him what he’s doing. “I’m looking for my keys”, he replies. “Where did you drop them?”, his mate asks. “Over there in the shadows”, he responds. “Then why are you looking here?” – “The light is better here” comes the reply.

² Mendonca, M. G.; Basili, V. R.; Bhandari, I. S., and Dawson, J. An approach to improving existing measurement frameworks. IBM Systems Journal. 1998a; 37(4).

³ Mendonca, Manoel G. and Basili, Victor R. Validation of an Approach for Improving Existing Measurement Frameworks. IEEE Transactions on Software Engineering. 2000 Jun; 26(6):494-499.

The method uses the goals of the user group to identify a set of *objects of study* and their associated *entities and attributes*. The *attributes* are then mapped to existing metrics and "inconsistent, missing and extraneous metrics" are identified. In a complementary analysis, legacy data is mined to identify "unexpected and useful information" about the *objects of study*.

A trial of the methodology targeted the information needs of a number of user groups in an IBM Canada Software Solutions Laboratory. They found seventeen groups in the laboratory whose tasks involved a need for customer satisfaction data. Of these seventeen, they selected three groups for a GQM-based analysis with a representative from each group participating. The method was effective in identifying missing metrics and providing input to future improvement cycles. The report does not address the method's efficiency, but there appear to be no economies of scale - six times the effort would have been required if all groups had participated. In addition, the method provides no information about the relative importance of the goals that could be used to focus improvement efforts.

The method proposed by Mendonca is rigorous and appeared effective, with the IBM laboratory making changes as result of the assessment. However, it seems resource intensive. In the trial, only one person from each of three functional areas (out of seventeen) was consulted about the goals of their area. Consulting more people in each functional area would have used more resources and may well have resulted in an explosion of goals and questions. It is also subject to the criticisms of GQM articulated by Card⁴ and Roche⁵, that

- Repeating a GQM exercise will not necessarily result in the same questions and metrics.
- GQM can be an open-ended exercise having no means to terminate the activities
- It can be impractical where the goals are not clear or where people are unaccustomed to thinking about and talking about their goals.
- Measurement feedback is required for the business process, the development process and the measurement process itself.

The authors concluded from the trial that the combination of top-down analysis using GQM and bottom-up analysis using data mining could form the basis for an integrated method of re-engineering measurement frameworks. The re-engineering paradigm, however, means a major commitment of time and resources at intervals. Many organisations may find it more politically and financially feasible to adopt the incremental improvement approach promoted by ISO/IEC standard 15939 – Software Measurement Process and the SEI's CMMI "Measurement and Analysis" process area.

About "Research Corner"

This is intended to become a regular feature in Metrics Matters. A piece of recent research will be reviewed in these pages. Contributions are solicited from both academics and industry. In addition, we will print any comments you want to make on these reviews.

⁴ Card, D. What makes for effective measurement? Editorial Comment, IEEE Software 1993

⁵ Roche, J. M.; Jackson, M., and Sheppard M. Improvement of the Goal-Question-Metric Method. Software Process Improvement '96; Brighton, U.K. 1996.

Sources of Metrics Related information

Industry Events

ACOSM 2002

Australian Conference on Software Measurement

(Hosted by : Australian Software Metrics Association)

“Making Metrics Work for Business”

Melbourne Australia - November 19th –22nd

Schedule for Submissions

Deliverable	<u>Due Date:</u>
Workshop and Conference Abstracts due:	May 31, 2002
Decision on acceptance:	August 31, 2002
Final Papers submitted for publication	October 31, 2002

Conference Schedule

Activity	<u>Delivery Date:</u>
Training Workshops:	November 19-20, 2002
IFPUG CFPS Exam	November 20, 2002
Conference:	November 21-22, 2002

Venue

Duxton Hotel
Flinders Street
Melbourne Australia

The Australian Software Metrics Association is holding its annual conference in November 19th - 22nd 2002 in Melbourne Australia. The first two days are workshop days and we are calling for vendors to submit their applications to run training courses at the conference. The emphasis will be on courses that deal with practical software measurement, process improvement and balanced scorecard and topics that deal with measurement from a business perspective.

Workshop presenters are asked to provide ASMA with a

- short description of their workshop, its target audience, duration, objectives, key topics and the agenda.
- A brief biography of the presenter and their experience

Proposals are required to be received by May 31st 2002. Decisions on acceptance will be made by August 31st 2002.

Workshop presenters will be paid in line with Australian Industry rates.

ACOSM is also asking for a call for papers for presentation at the conference - abstracts are due May 31st 2002. The topic for the conference is around Software Measurement and its usefulness to the business.

Abstracts and workshop proposals and requests for more information contact :
ASMA Administrator : Robyn Smith: asmavic@ozonline.com.au
Conference Convenor : Paula Holmberg : paulah@au1.ibm.com

We look forward to hearing from you.

IFPUG CFPC Test

IFPUG CERTIFIED FUNCTION POINT COUNTING TESTING

The next sitting for CFPC Accreditation will be held in Melbourne on

Monday 17th June 2002,

subject to approval from IFPUG.

For further details and application form go to the ASMA web site at www.asma.org.au.

or

Contact Robyn Smith on 03 9844 0560.

2002 Software Conferences

8th IEEE International Software Metrics Symposium

4 – 7th June 2002 at Ottawa, Canada

For more information: <http://www.software-metrics.org/>

International Conference on Software Quality

28 – 30th October 2002 at Ottawa, Canada

For more information: <http://www.icsq.org>

Metrics related Information Sources

<h2>Reference Books</h2>

These book reviews were extracted from: <http://www.gifpa.co.uk/>

Establishing A Company-Wide Program

Robert B. Grady and Deborah L. Caswell

Prentice Hall - ISBN 0-13-821844-7

In his Foreward, Tom DeMarco of The Atlantic Systems Guild writes, "...Fortunately there are at least a few examples of sensible approaches to measurement, and one of the most compelling of these is described in *Software Metrics: Establishing A Company-Wide Program*. This book tells of one company's need for a measurable, controllable software process and of the very professional effort the company mounted to meet that need. If the effort had been less ambitious, this book would have ended its days as a long memo, of interest only within the author's company. But the approach was so broad and so multi-faceted that it covers most of the ground necessary to set up such a program in any company. It tells of the metrics chosen, the tools used to collect and digest them, the selling job to get people involved, the forms, the training sequences, the documentation, the results and the costs.

"Bob Grady and Debbie Caswell were not just passive observers of the program they describe, they were right there in the middle of it. Their experience is germane, their advice is informed and heartfelt, and their enthusiasm is catching. The story they tell is useful preparation for anyone setting out to collect and use software metrics.

A practitioner's guide to improved product development

K.H. Möller and D.J. Paulish

Chapman & Hall ISBN 0-412-45900-0

Software metrics techniques are increasingly being used by organizations wishing to improve the quality of the software they develop and the efficiency with which it is produced. This book describes 'best practices' and will show readers how companies in Europe and North America are implementing successful software metrics programmes.

The authors present the material in the form of case studies supported by tutorial materials which will be extremely useful to individuals introducing a metrics programme into software project management as well as to those wishing to improve on existing metrics programmes.

Materials in this book have been developed from work carried out by the PYRAMID research project, a collaborative research project funded by the European Commission under the ESPRIT II programme.

Karl Heinrich Möller is Technical Director and Daniel J. Paulish is the Task Leader for the ESPRIT PYRAMID project, Siemens AG, München, Germany.

Interesting web sites

<http://www.standards.com.au>

Standards Australia Home Page

<http://www.iso.ch>

International Organisation for Standardisation

<http://www.ieee.org>

Institute Electrical & Electronics Engineers

<http://www.ozemail.com.au/~sqain>

SQA Software Quality Assurance NSW

<http://www.cosmicon.com>

Cosmic-FFP web page

Metrics Matters web site of the month:

Do you know what SPICE is?

It stands for **S**oftware **P**rocess **I**mprovement and **C**apability **d**etermination. More information can be obtained from

<http://www-sqi.cit.gu.edu.au/spice/>

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.asma.org.au)

[HTTP://WWW.ISBSG.ORG.AU](http://www.isbsg.org.au)

[HTTP://WWW.BANNISTER.COM/IFPUG/HOME/DOCS/IFPUGHOME.HTML](http://www.bannister.com/ifpug/home/docs/ifpughome.html)

STATE CHAPTERS - ALPHABETICALLY

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 Meetings/Presentations are normally held on the fourth Monday of every month, commencing 5.30pm. Please refer to the ASMA web site for further details or contact Maria Patane

ASMA NSW 2002 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	Mike Berry	Michael.Berry@student.unsw.edu.au	9712 1605

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

In the first week of March the ASMA (Victoria) held its Annual General Meeting and committee elections.

On behalf of us all at ASMA we thank those leaving the committee particularly Ms Robyn Lawrie for her work on the ASMA executive over the past ten years and Mr Paul Radford for his period of Presidency. We are pleased to welcome two new members to the executive team John McCrae from *McCraeIT* and Bhaskar Majumdar from *Deloitte Consulting*.

In 2002 , ASMA (VIC) will increase the number of its free evening seminars. Our schedule for these seminars from now until November is published below. This year we are planning to

upgrade our WWW site to provide more resources including a means by which members can swap ideas and experiences on specific topics. We are currently in the process of planning ACOSM 2002 to be held again in Melbourne November 2002 and would welcome any assistance from our members to participate in the organising committee.

Wednesday 22 MayPresenter: Tim Hastings Topic: Is Software Development Estimation Possible?
Tuesday 25 June
Wednesday 24 July
Tuesday 27 August
Wednesday 25 September
Tuesday 29 October
Wednesday 27 November

ASMA (Vic) Committee 2002

POSITION	NAME	Company	EMAIL	WORK
President	Pam Morris	Total Metrics	Pam.Morris@totalmetrics.com	03 9882 7611
Public Officer	John Ogilvie	IBM GSA	mmaogil@au1.ibm.com	03 8605 0505
Treasurer	Robyn Smith	Trearc Pty Ltd	asmavic@ozonline.com.au	03 9844 0560
Program Director	Michael Stringer	Sage Technology	m.stringer@sagecomp.com.au	03 9888 1760
Committee Member	Bhaskar Majumdar	Deloitte Consulting Outsourcing	bhmajumdar@dc.com	03 615 7672
Committee Member	Paula Holmberg	IBM GSA	paulah@au1.ibm.com	03 8605 0562
Committee Member	John McRae	McRae IT	John.mcrae@mcraeit.com.au	03 9882 2060
Committee Member	Terry Wright	Multimedia Victoria	terry.wright@mmv.vic.gov.au	03 9651 9006

.....end of Metrics Matters.....