

ASMA/SQA (NSW)

Australian Software Metrics Association / Software Quality Association (NSW)

ABN 86 541 712 427

Metrics Matters

E-JOURNAL OF THE AUSTRALIAN SOFTWARE METRICS ASSOCIATION / SOFTWARE
QUALITY ASSOCIATION (NSW)

September, 2005

**© The e-Journal of the Australian Software Metrics Association / Software
Quality Association (NSW), circulated to members Australia wide.**

ASMA/SQA (NSW)

www.asma-sqa-nsw.org.au

e-mail:

info@asma-sqa-nsw.org.au

The Publisher

e-mail:

info@asma-sqa-nsw.org.au

Welcome



Mark Exall

This issue marks the first edition of the Metrics Matters e-Journal under the ASMA/SQA (NSW) banner.

As a result, I have invited the chairman of ASMA / SQA (NSW), Helen Aitken to say a few words. She shares with us what software metrics activities are happening in NSW and give news on matters that affect us on a national scale. Read her musings and get a sense of the direction that metrics in Australia is heading in, as we find out what questions are on Helen's mind. Also, get a feel for the type of person Helen is, as she adds her own personal touch through her conversational style.

In this issue we also take a sneak preview of the Australian Conference on Software Metrics (ACOSM 2005). It is hard to believe that it is that time of year already!

We welcome a submission from Greg Gilbert on the subject of military software within the Australian Department of Defence. He outlines some of the challenges faced within Defence, such as the issues that need to be addressed when considering commercial off the shelf software. Also, he covers the extra demands that are placed on software due to the Defence environment, which add some food for thought. For those interested in other software measurement methodologies, Greg examines Use Case Points (UCP) as an alternative approach to the traditional function point methodology.

Finally, in Last Words, Ted Smillie follows on from his editorial in the August ASMA/SQA NSW newsletter which asked whether the article attacking the reliability of lines of code and function points for software sizing is obscuring more important quality metrics. Here, in "IT Spending – A Necessary Evil?" he goes further and asks whether we are neglecting an even more important measurement – benefits realisation. Thanks Ted for challenging our thinking once again.

I would like to give Sophia Poulos a special thanks for her general guidance and as joint editor. You helped me hang in there!

Feedback on either the contents or the format of the Metrics Matters e-Journal is always welcome. Please email comments to:

info@asma-sqa-nsw.org.au.

Best Regards,

Mark Exall

Editor/Publisher

Contents

ASMA / SQA (NSW) Chairmans's	
Musings	3
A Sneak Preview of ACOSM 2005	6
Estimating Military Software	9
ASMA/SQA (NSW) Contacts	18
About Metrics Matters	18
Contributions...	18
Last Words...	19



ASMA / SQA (NSW) Chairman's Musings

By Helen Aitken

Mark asked me to write an article about what we in NSW have been up to.

Last year, after many years of holding combined meetings at least once or twice a year, we decided to amalgamate ASMA (NSW) and SQA (NSW). We thought it would make life easier to combine meetings, membership lists and administration. You are welcome to request a copy of our business plan by e-mailing info@asma-sqa-nsw.org.au. Little did we know that we would evolve to be the only "ASMA" in Australia.

SQA (NSW) has had a policy of holding a membership meeting each month and ASMA (NSW) has aimed for similar membership meetings, although not every month. We have a website (now www.asma-sqa-nsw.org.au) where we publicise these meetings and request that speakers give us slides of their talk. We also have old copies of our newsletter and a few copies of Metrics Matters. All this has been running fairly smoothly, although extra committee volunteers would be very welcome.

Last year Professor Ross Jeffrey, one of the founders of ASMA (NSW), asked if we would like to hold a combined ISESE / Australian Conference on Software Metrics (ACOSM) conference. Ross was responsible for the very successful IEEE Metrics / ACOSM (2003) partnership. This year's conference is being held in Noosa – a nice change from Sydney or Melbourne but a bit of a stretch for a geographically dispersed organising committee.

With the disappearance of ASMA (National) we have been asking questions about our own corporate governance issues. The answers have been "interesting" to say the least and more work is being done – both at the local and national level. We have been assisted by the ACS National Technical Chairman, Professor Karl Reed (La Trobe) and Barry Wong, the ACS (NSW) branch manager. Most of the ACOSM organising committee has taken on a second role of working out where we are going. SQA is a National SIG of the ACS so we have also been talking with SQA allied SIGs in other states. The image of a swan – stately on top of the water but pedalling like mad underneath – comes to mind. Ted Smillie (ts@mrose.com.au) is chairing these initiatives.

Questions that have been asked include:

- What is our mission? Why do we exist and what do we hope to achieve?
- What should we call ourselves? SQA (NSW) has been joking for years that we'll combine ourselves with ASMA and then change our joint name to SPIN (the Sydney SPIN group was created by my husband, Doug Jay).
- How do we know that we've got there (measurements ☺)?
- Who are our members? ASMA / SQA (NSW) is NSW based but we have recently accepted a number of members from out of state – this is outside our charter as currently stated on our website.
- Who gets Metrics Matters?
- What is the future of ACOSM?
- Currently ASMA / SQA (NSW) is incorporated under the ACS rules as part of the National SQA

SIG. We asked to see the documentation and there is none. Should we each incorporate under our own state regulations? What is the cost of that and what are the benefits?

- What about members, for example IBM, who have offices in many states?
- How do we interact with ISBSGs? (www.isbsg.org.au) Currently ASMA / SQA (NSW) is the Australian member with Pam Morris of Total Metrics as our delegate.

On a personal basis, this year has also been interesting. My first child, Alexander, was born in January and attended his first teleconference when he was 5 days old. We look forward to catching up with you at Noosa.

Helen Aitken
ASMA / SQA (NSW) chairman
e-mail: info@asma-sqa-nsw.org.au
web: www.asma-sqa-nsw.org.au

⌘

Putting the fun back into
Function Point Counting
(seriously)

SCOPE
Project Sizing Software™

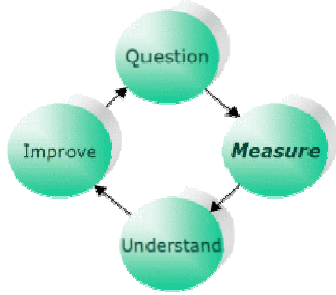
- Counts IFPUG 4.2 Function Points
- Provides graphical FP audit trail of changes to Functional Requirements and rework
- Historical record of software application and project sizes with 30 different reports - online or printed
- Ask about our **50% discount** for crossgrades from Function Point WORKBENCH™

FREE Evaluation Copy - contact Total Metrics : admin@totalmetrics.com
WWW.TOTALMETRICS.COM



EARLY BIRD Registration Now Open! Book Before 20 October

ACOSM 2005



AUSTRALIAN CONFERENCE ON SOFTWARE
MEASUREMENT 2005

THURSDAY - FRIDAY

17 - 18 NOVEMBER

-

WORKSHOPS

TUESDAY - WEDNESDAY

15 - 16 NOVEMBER

-

SHERATON NOOSA HEADS

NOOSA HEADS QUEENSLAND AUSTRALIA

<http://www.asma-sqa-nsw.org.au/acosm>

Question.

Are requirements complete?
Is scope under control?
What's the development effort?
What's the return on investment?

Measure.

Function point analysis
Maintenance & support effort
Project performance
Metrics strategies

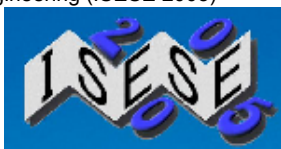
Understand.

ISBSG & benchmarking
Changing requirements
Functionality
Productivity

Improve.

Estimation
Software quality
Change management
Process maturity

Co-located with the 4th International
Symposium on Empirical Software
Engineering (ISESE 2005)



Special Invitation

ACOSM 2005 is the Asia-Pacific
region's premier software
measurement event.

Since 1991, ACOSM has been a
significant annual event for the software
measurement, software quality assurance
and process improvement industry.

Topics for discussion centre on the theme of software
measurement as a key component in a process improvement
framework. Sessions focus on the innovative and practical use
of measurement to provide true benefits to software suppliers
and users. Speakers come from commercial, academic and
government organisations Australia-wide and International.

A key presenter at the conference is the inimitable **Barry
Boehm**, who will be joining us for a presentation on the
"Future of Software and Systems Engineering Processes". His
contributions to the field include the Constructive Cost Model
(COCOMO), the Spiral Model of the software process, and the
Theory W (win-win) approach to software management and
requirements determination.

In addition, **workshops** are run in the two days leading
up to the conference. Pam Morris from Total Metrics will be
holding the COSMIC (intensive) one day workshop held for the
first time in Australia. Find out about the IFPUG Software
Metrics Specialist Certification from David Cleary, Charismatek
and more.

Also at ACOSM 2005, the only opportunity to sit the **IFPUG
CFPS exam** in Australia this year.

Three top reasons why you should attend the ACOSM Conference this year

1. Learn how Australian commercial and government
organisations are forging ahead using the strengths
of measurement techniques; understand the
challenges and opportunities of measurement
frameworks for you and your organisation
2. Great networking opportunities with other delegates
3. Realistically priced at just \$743 (includes GST) for
ASMA/SQA members (but only before 20th October)

.... and the opportunity to see new research by speakers from
the International Symposium on Empirical Software Engineering
(ISESE) conference conveniently co-located with ACOSM (**for
no extra cost**).

Conference and workshop places are limited
<http://www.asma-sqa-nsw.org.au/acosm>

A Sneak Preview of ACOSM 2005

By Mark Exall

As the only conference in Australia devoted to software metrics, I know that for many software metrics practitioners who want to keep up with the latest developments, ACOSM is a major highlight for the year. So what can we expect from this year's conference?

Here is a taste of what is in store...

◆

No doubt one of the highlights will be key speaker, Dr Barry Boehm, who will be presenting on the "Future of Software and Systems Engineering Processes". Discover what are eight relatively surprise-free trends and two "wild card" trends in software and systems engineering. Explore the likely influences of these trends on software and systems engineering processes between now and 2025. And learn about an emerging three-team adaptive process model for coping with the resulting challenges and opportunities of developing 21st century software-intensive systems.

◆

Emilia Mendes and Chris Lokan will speak on "Cross-company and Within-company Effort Estimation Models using the ISBSG Database". Delve into the International Software Benchmarking Standards Group (ISBSG) database and find out how successful a cross-company model is at estimating effort for projects from a single company when the model is built from a data set that does not

include that company. Then learn how successful a cross-company model is, compared to a within-company model.

◆

Want to know how you can better manage and thus reduce software project risk by using the concept of the "sum of the new"? Jonathan O'Donnell-Young will speak about "Promises, Promises - the Art of Committing to the Possible". The good news is there is a role for software metrics. Attend this session and you will find out why.

◆

Alan Abrahams will cover "Accounting for Profitability and Productivity using Function Points". The spotlight will be on a high-level executive reporting system called productivity accounting. In short, it can show whether company productivity is increasing or not and the impact that this is having on bottom line results. Obviously, there is a focus on function points to make this valuable, informative and relevant. Further, a case study is used to show it all in action.

◆

In addition to many well-respected speakers, ACOSM 2005 will be a great opportunity to network with other likeminded people. There are also workshops to attend, the CFPS exam to sit and the International Symposium on Empirical Software Engineering (ISESE 2005) sessions to broaden our world of understanding.

See you there!

⌘

Faster, Easier, Smarter!



Function Point **WORKBENCH** 6.0

PATENT PENDING

Redeveloped
for Windows XP

Revolutionise your Function Point Analysis and your Project Estimating with the new **Automated Counting** and **Early Life Cycle Estimation** capabilities.

- Estimate Time, Effort and Cost
- Agree Functional Requirements
- Establish a Scope Baseline
- Monitor Project Size and Scope
- Forecast and Assess Support Resources
- Benchmark and Manage Your Applications Portfolio



CHARISMATEK
SOFTWARE METRICS

CHARISMATEK Software Metrics - www.charismatek.com
Aligning the Software Development Process with Your Business Imperatives

Special Reports Available from ISBSG

ASMA Members Special Offer

Over the last 12 months the ISBSG has produced a series of Special Reports. These reports are primarily aimed at IT practitioners featuring topics and techniques to help them in their day to day work.

The following Special Reports are available to ASMA/SQA members at a discount. Go to www.isbsg.org, select Purchase Products, then On-line Order Form. Simply select "ASMA" from the drop down list box to receive a 5% discount off the normal price of \$25.

Early Lifecycle Software Estimation

Often, when a software project is simply an idea, you need to provide an indicative estimate of the possible cost and duration to provide an indication of whether the project idea is feasible. This paper describes an early lifecycle estimating technique that uses a combination of formulae and the ISBSG project delivery rate tables. Using the information in this paper you can produce an estimate range for the effort and duration of a proposed software development. Practical examples and the required tables are provided.

Team Size impacts development productivity and delivery

The ISBSG data shows that there are three main factors that impact software development productivity: programming language, development platform and team size. The first two have the most significant effect but it is also important to consider the impact of team size. The latest ISBSG Special Report reveals that teams of nine or more are significantly less

productive than smaller teams. Project managers faced with larger teams should adjust their project estimates to reflect this lower productivity expectation.

Software Project Costs

In this report we look at costs per function point and costs per hour, and provide some useful rules of thumb.

Package Based Projects

This ISBSG special analysis reveals that choosing and implementing a package has advantages over developing new software, provided that the implementation is either turnkey or utilises customisation facilities provided with the package. Package projects that involve changes to the package source code perform worse than development projects.

Estimates - how accurate are they?

We look at estimates of size, effort, duration and cost; how people have gone about estimating their projects; the accuracy of the estimates and the relationships between estimates. Here are some of the findings:

- Size estimates are usually based on a data model; functional specification or analogy with a previous project
- Project effort estimates are only accurate for less than a quarter of projects
- Despite effort being poorly estimated 51% of projects were delivered on time
- When functional size-based techniques are used for a cost estimate, the estimate is within 20% of the actual cost 90% of the time [Full report](#)

⌘



Estimating Military Software

The Problem Facing the Defence Project Office

By Greg Gilbert

Abstract

The Australian Department of Defence outsources the development of its mission systems. These systems increasingly incorporate integration of Commercial-Off-The-Shelf (COTS) technologies and *black box software development* which customises and extends the technology through its Application Programming Interface (API). Defence outsourcing strategy seeks a competitive fixed price proposal to deliver mission systems. But, for a variety of reasons, delivery regularly requires several phased iterations in the form of contract extensions. Market testing (seeking tenders) and changing the Systems Integrator for an iterative development phase, is obviously antagonistic to system development. For this reason, Defence Systems Integrators benefit from uncontested contract extensions. The challenge facing a Defence Project Office is how to budget for and judge the commercial value of a sole-sourced contract extension. Making matters worse, the Defence trend to *black box software development* compromises the relevance of the traditional Function Point methods. Use Case Points (UCP) offer promise and could provide an estimate early in the specifications phase, except that more effort is needed to mature UCP as an appliance. Their implementation also hints at changes to current Defence Project Office structure and methods.

Introduction

The Australian Defence Force outsources the development of its

mission systems. Current outsourcing strategy seeks a fixed price proposal to deliver a described outcome. But changing and unknown requirements mean that specifications are frequently incomplete at the time of contract. For this reason the Australian Defence Force supports iterative and evolutionary development models which allow the gradual building of a system.

Each iterative phase is customarily separately priced, delivered and managed via a scope change to the existing contract. Clearly it benefits both the Defence buyer (Defence Project Office) and the supplier (Systems Integrator) to maintain partnership confidence for the duration of the build, as replacing the Systems Integrator in a complex, partially completed development poses both an obvious cost and risk. Retaining the same Systems Integrator throughout several phases of build requires that a Defence Project Office is able to reconcile value for money for each iterative phase of development.

The challenge facing a Defence Project Office is to know when it is getting fair value without the expense, delay, disruption and potential risk of market testing each iterative phase of development.

The problem is not new. Military software projects are more expensive than their commercial counterparts. They are characterised by an extensive suite of specifications constraining both method and design. In "Software Assessment, Benchmarks and Best Practices", Capers Jones [1] differentiated military software from

other software developments by the scale of its oversight documentation, and by the slow uptake of commercial methodologies and practices. In particular, Jones lamented the military's lack of application of function point metrics to the software estimation problem.

To some extent this may be due to the military's lack of application of function point metrics, the limitations of function points themselves or to the suitability of the parametric estimation models to assist a Defence Project Office prepare a software development budget. Whatever is the case, budgeting and reconciling value for money of outsourced Military software development is an important and difficult problem. There is a clear need for methods and tools that assist a Defence Project office to set and manage the software development budgets and to quantify the value for money of sole sourced phased extensions.

Military Software Development

Jones mapped six broad categories in which software projects share similar attributes but he differentiated military software as a category in its own right due to its "...unifying theme..." of "...adherence to standards...such as DoD standards 498 or 2167A"[1]. Jones explained that one of the reasons for separately classifying military software is that during the 1970s through the early 1990s military software had developed a unique set of software documentation criteria "...approximately three times larger than civilian norms..." making "...military software project productivity the lowest in any industry..."[1].

Outsourced military software developments are further subjected to legal oversight designed to penalise

Systems Integrators if the project is late, incomplete or fails acceptance. Jones identified this form of military software development where the implications of military standards and potential litigation make such projects "...A fairly unique subtype..."[1].

It is fair to speculate that the additional legal oversight associated with outsourced military software developments adds both costs for the administration of the contract and the risk of penalties which might apply. On top of this, fixed price outsourced software development incurs an estimation risk component making it then by far the most expensive of any form of software development.

For this reason, Defence and industry need to come to terms with techniques to quantify the cost of fixed price outsourced software development. Failure to address this issue leaves Defence project managers wondering why it cost so much, and conversely, the Systems Integrators continually on the back foot attempting to justify it. This business environment creates the potential for friction, suspicion and mistrust.

The Move to Civilian Practices

Jones observed that:

"...In recent years, military software applications have been moving towards civilian practices and away from the unique criteria contained in the old Department of Defense standards. However, old practices die hard, and military projects still tend to be more concerned with oversight ...than civilian projects..."[1].

This may be understandable given Defence's long outsourcing history. However, the move to less costly civilian practices may also be stymied behind a cultural barrier held all the more higher by the mistrust built up by Defence and industry failing to come to terms with techniques to quantify the software development cost.

For a long time Defence shunned COTS technologies as not being sufficiently robust for the battlefield. For example, Defence information security requirements are different from commerce. Where a commercial software vendor might adopt the approach to build, sell and fix security faults as they are found, Defence forces in general cannot afford to field systems with security flaws as the consequences are too grave. While the cost of these requirements is hugely burdensome on Defence budgets and impact the time to deliver solutions, they also help explain Defence's initial reluctance to engage COTS technologies.

Budget and Schedule Driven Change

The burdensome cost and time to delivery factors have forced changes to the Defence software development environment. It is now more than 10 years since the U.S. Secretary of Defense, William Perry, issued a directive that the military services should no longer regard Department of Defense (DoD) standards as being mandatory. The DoD community was to adopt civilian best practices.

In the intervening period, the US DoD, like commerce, has embraced Component Based Software Development (CBSD) in an effort to reduce the cost of development and speed up the time to deployment. The

Australian Defence Force has followed suite simply because much of its mission systems technology is acquired from US DoD. However, not unlike their commercial counterparts Defence Project Managers are under time and budget pressure to field systems that give their forces a "competitive edge" and CBSD is a key strategy in delivering solutions more efficiently.

COTS Component Integration

At the heart of CBSD is an assumption that parts of large software systems reappear with such regularity that these parts should be written once and re-used many times over. Perhaps not surprisingly COTS examples include commercial database technology, operating systems, Java Object suites, CORBA, COM, .NET and others. However, in Defence, CBSD is driving the integration of COTS technologies into the combat critical mission systems. The US DoD is moving inexorably forwards to the full engagement of COTS technology in the delivery of bespoke mission systems. In the 1999 the Defense Authorization Bill, Congress directed that future versions of the US Department of Defense's Joint Mapping Toolkit (JMTK) be based on commercial technology. The C/JMTK (Commercial Joint Mapping Toolkit) introduced in 2003 exemplifies the application of commercial technologies and software development methodologies to the US Department of Defense. Its website proclaims:

"The C/JMTK program is particularly challenging because it fundamentally changes the way the US Department of Defence conducts business in several ways. First, the C/JMTK is a major acquisition to provide a common commercial software

toolkit for geo-processing and visualisation to the DoD. Second, the C/JMTK moves the DoD from a functionally-oriented way of organizing architectures and systems into an object-oriented way of organizing architectures and systems. Third, the C/JMTK provides for a horizontally-integrated, services-based architecture to facilitate the DoD-wide interoperability. Successful execution of the C/JMTK Program requires preparing the Community for a major migration toward a new programmatic and technical way of deploying C2I Mission Applications within the DoD.”[3]

The significance of this is twofold. Firstly, the Australian Defence Force will follow suit. The alternative is simply unaffordable. Secondly, the COTS technology vendors noting the trend and eyeing DoD mission systems opportunities are developing and promoting Standards based COTS technologies and with fine grained APIs. Standards based COTS technologies promise that integration of product from different vendors is both simplified and assured and the fine grained APIs providing additional design and implementation flexibility. These factors are a further enticement for Systems Integrators to engage COTS technologies in Defence solutions.

COTS integration in Defence mission systems means that the Systems Integrators and developers no longer have access to the source code and instead are relying on a vendor’s executable objects. Despite this attendant risk, as of February 2004, more than 130 USA DoD mission programs are migrating or developing using the C/JMTK [3].

Software Estimation

Perhaps two questions face the software estimation industry:

- Firstly, how well do the current software estimation tools deal with the universal trend to COTS integration and *black box software development*; and
- Secondly, how well do the current software estimation methods and models support a Defence Project Office outsourcing development of its mission system?

Regrettably perhaps, there is little evidence, in Australian Defence mission systems development circles, of any uptake of function point software estimation methods. There is also little evidence or interest in the uptake of the parametric estimation tools such as COConstructive COSt Model (COCOMO II) or SEER-SEM.

This lack of interest and application of these software estimation methods, and their contemporary parametric estimation models, may be attributed to:

- the origins of function points themselves;
- inapplicability of the function point estimation methods to *black (and grey) box software development*; and
- failure of the parametric estimating tools to assist the Defence Project Office budget process.

Origins of Function Points

Function point estimation has its genesis in the Information Systems, Management Information Systems (MIS) software category. Traditionally, most widely used by those programming in the administrative, commercial and MIS arena rather than by developers working on real-time, embedded systems or those programming systems with a high algorithmic content. Military mission systems software development is different to MIS software. The time critical nature of military information, the message based information exchange protocols designed to overcome the austere bandwidth, processing and storage environment, and an extensive collection of coalition interoperability software development constraints promote a significantly different development environment than that of a file based business MIS significantly challenging the adequacy of function point estimation.

Black Box Software Development

A telling factor on the suitability of function point based software estimation is that it is not applicable to the COTS component integration. This is because the function point estimation at the core of the parametric models is based on development from scratch in the case of COCOMO or, development from scratch plus reuse of existing code for COCOMO II. History has already revealed the impact on these models caused by the evolution in software development. COCOMO was originally published in 1981 and reincarnated as COCOMO II in 1995 to deal with white box software adaptation which exploits code reuse and adaptation. It is no mute point, but in grey box and black box COTS integration the components are

not adapted but used as is. This emasculates the application of function point estimation and its associated parametric software estimation models from estimating COTS integration.

Defence Project Offices

Australian Defence mission systems project offices and acquisition organizations are staffed with a mix of uniformed personnel and public servants. In general, the uniformed personnel assist in defining, justifying and delivering the capability into service, whereas the public servants manage the accountability elements. Missing from the mix of project office staff is the IS business analyst. Consequently Defence Project Offices are approaching the market with broadly defined functional requirements, arguably inadequate software specifications and an unconvincing software budgeting process. The latter lacking both tools and trained staff. To some extent this is the fault of the tools themselves. While function points are a tool in the hands of the IS business analyst, the parametric models are tools for Systems Integrators. This becomes more evident when considering the utility to Defence Project Offices of the new wave of COTS integration parametric estimation tools.

COTS Parametric Estimating Models

The impact of COTS component integration is not lost on the estimation industry leaders. In their paper Abts, Boehm and Clark [4] identified the limit of COCOMO II and the ever more prevalent trend to grey box and black box COTS component integration. Their paper discussed a model to extend COCOMO II to predict "...lifecycle costs of using COTS component...".

They also provided an overview of the COCOTS model and its limitations. Specifically, they recognised that application of any predictive model such as COCOTS is contingent upon empirical data. They observed that acquiring such data is difficult, costly and time consuming. In view of this they proposed:

“...a multi-step modeling methodology incorporating Bayesian statistical techniques (which allow the establishment of) initial model parameters values based on a blending of numbers derived from expert judgment, with calibration numbers derived from whatever empirical data is at hand...”[4].

Limitations of the Parametric Estimation Models

SEER-SEM is similar to COCOTS without, however, the initial seeding of COCOTS estimation parameters. Like COCOTS, COCOMO, and other parametric estimation models SEER-SEM provides for model calibration based on empirical project statistics. The parametric estimation models are evidently tools for the Systems Integrator. After all, only the Systems Integrator has the capacity to capture the empirical data to drive the model. Without empirical data these estimation models hold little value in the hands of a Defence Project Office attempting to estimate a budget or reconcile the value of a sole sourced proposal. Models like COCOTS have other limitations including the initial technical analysis phase necessary prior to forming an estimation effort. Defence Project Offices do not have the skills to undertake the initial analysis required by the COCOTS method.

What is Needed

What Defence Project Offices need is methodology to address budget and value for money for COTS component integration projects. Ideally, Defence Project Offices need to be able to get access to early software estimations to at least set a budget. A sole sourced fixed price proposal that was within 30 percent of a budget estimate would be good enough, right now, to argue value for money. What would be better of course is if the continual application of such a methodology resulted in improved budget forecasts.

Methods that would work in the military world have to have the following characteristics:

- be applicable to real-time and near real-time software developments;
- be independent of requiring a detailed knowledge of the COTS component involved (i.e. focus on the what not the how); and
- be applicable at the requirements or specifications phase.

Use Case Points

Use Case Points (UCP) is a relatively new software estimating model which might serve this purpose. UCP were initially described by Karner [5] are like function points but are derived directly from use case models. Use case models are increasingly being used to describe functional requirements and are becoming commonplace for object oriented developments, COTS integrations, and are core to the Rational Unified Process.

The principal advantage of UCP is that they are available early on in the project; even before function point estimation might be entertained. Indeed some researchers have focused on using UCP to derive function point estimates of software. This “cross firing” of UCP to function points perhaps providing the opportunity to apply empirical function point cost factors collected over many projects.

Some work has been done on UCP but little is reported by way of practical experience in the application of the estimation methodology. Although one case study [10] on the application of UCP to a web development reported the variance between actual effort and estimated effort to be as little as 15 percent. And, Damodaran and Washigton [7] compared UCP with function point estimation and concluded that:

“...use case points method of effort estimation is a very valuable addition to the tools available to the project manager. The methods can be very reliable or just as reliable as other effort estimation tools such as COCOMO, function point and lines of code...and with standardisation...have the potential to become a mature and widely accepted estimation tool...” [7].

But, they also observed:

“...Industry use of Use Case Points method of estimation is very rare...One possible reason (they speculated) why there has not been more widespread use of this method in industry is that this method has not been incorporated in the popular project estimation tools.

Another reason may be the relative newness of use case points, even use cases as a standard method of describing requirements it is not possible to use them to estimate effort and cost...”[7]

This latter view is supported by Smith [8] who found several difficulties, such as lack of consistency in use case specification styles and formalities, preventing application of use cases to estimating effort.

Others have invested effort into formalizing uses cases [9]. However, what is missing is research addressing the qualities of effort estimations obtained from the formalised Use Case models.

Norwegian researchers at University of OSLO and University of Science and Technology are active and offer some current research in the field of UCP including case studies from industry [6] and have completed some initial analysis on the application of UCP to incremental large-scale development [11]. In the first instance the researchers analysed three small projects and the effectiveness of applying UCP to estimate effort. While they concluded that UCP can be used for successfully estimating effort they also noted that the “design of use case models has a strong impact on the effort estimates” [6], reinforcing the need for some research on the effort forecasting qualities from formalised use case models. In the more recent work researchers describe:

“ ...An industrial study of effort estimation based on use cases, the UCP method. (In this case,) The original method was adapted to incremental development and evaluated on a large industrial system with modifications from the previous release. (And) ...

The method was calibrated using data from one release and it produced an effort estimate for the successive release that was 17% lower than actual effort.”[11]

These initial studies are early indicators that UCP might be applicable and in scenarios relevant to the defence context. However, from the research cited above, it seems likely the quality of UCP estimation is closely aligned to the qualities of the use case model. This suggests that research is needed to explore this relationship. Further research may also be needed to validate the UCP estimation methods in real-time and complex computing environments. The latter, an important issue for a military software estimating tool.

Impact On Defence

Assuming that formalised use case modelling provides a platform suitable for UCP estimation, more work is required to turn this into a budget estimate. UCP don't estimate cost alone, they offer the potential to estimate development effort. Development effort is a principle cost driver in software development projects. And, there is a substantial body of work [1] [4] relating development effort to software development project costs which would need to be applied to derive a budget estimate. Some researchers have found ways to “cross fire” UCP into function points (ie convert UCP to function points) to take advantage of the established budget modelling parameters of function points. This is a reasonable short term approach that might be applied by Defence in the absence of UCP derived empirical cost data.

Even so, key to estimating software development effort is the adoption of use cases as an integral part of the functional requirement specification. Use cases for software development are not something that Defence has previously produced. So there is a skill shortage to be addressed.

Use cases are a more refined form of functional specification than that currently applied to Defence specifications. This is a problem of its own right as Defence Project offices and the IT Industry in general have struggled to provide adequate software requirements definition. In Defence Project Offices the problem may be exacerbated due to a systemic failure to engage business analysts even in its IT intensive Project Offices.

Defence fixed price outsourced software development contracts are driving these issues to a head. To fix the price a Systems Integrator needs to now know with far greater precision the things the software must do. Defence Project Offices are now challenged to provide this definition. IT business analysts are instrumental to achieving this and in doing so provide the platform for the introduction of UCP driven software estimation.

References

1. Capers Jones (2000): Software Assessments, Benchmarks and Best Practices, ISBN:0-201-48542-7, Addison Wesley April 2000.
2. Capers Jones, Applied Software Measurement - Assuring Productivity and Quality (2nd Edition), ISBN 0-07-032826-9, McGraw-Hill 1996.
3. www.cjmtk.com (Commercial Joint Mapping Toolkit), Sponsored by USA National Geospatial-Intelligence Agency, Reston Virginia.
4. Abts, C., Boehm, B. and Clark, E., "COCOTS: A COTS Software Integration Lifecycle Cost Model – Model Overview and Preliminary Data Collection Findings", 2000, USC Centre for Software Engineering <http://sunset.usc.edu/publications/TECHRPTS/2000/usccse2000-501/usccse2000-501.pdf>.
5. Karner, G., "Metrics for Objectory", Diploma thesis, University of Linköping, Sweden No. LitTH-IDA-Ex-9344:21. December 1993
6. Ande, Bente, D. Dreiem, Dag, Sjøberg, and Magne Jørgensen, Estimating Software Development Effort Based on Use Cases – Experiences from Industry", 2001, M. Gogolla, C Kobryn (Eds): UNL 2001 – The Unified Modelling Language. Modelling Language Concepts and Tools, 4th International Conference, Toronto, CANADA, October 1-5, 2001, LNCS 2185 Springer
7. Damodaran Mel, Washington Aqua Netta E., "Estimating Using Use Case Points", Computer Science Program, University of Houston-Victoria, Vitoria, TEXAS 77901, USA, In The Proceedings of ISECON 2002, v 19, San Antonio ISSN: 1542-7382
8. Smith, John, "The Estimation of effort based on use cases", Rational Software. Cupertino, CALIFORNIA, TP-171, October 1999
9. Graham, Ian, "Migrating to Object Technology", Addison-Wesley, 1995.
10. Argawal Rakesh, Santanu Banerjee and Bhaskar Ghosh, "Estimating Internet Based Projects: A Case Study", Quality Week 2001 – The Internet Wave, paper QW2001 Paper 6W2, 2001
11. Parastoo Mohagheghi, Bente Anda and Reider Conradi, "Effort Estimation of Use Cases for Incremental Large-Scale Software Development", ICSE'05, May15-21, 2005, St Louis Missouri, USA, 2005, ACM 1-58113-963-2/5/2005

⌘

ASMA/SQA (NSW) Contacts



ASMA/SQA (NSW) - for membership enquiries: Chairman: Helen Aitken Administration Support: Jeff Lock	Email: info@asma-sqa-nsw.org.au
Postal Address	ASMA/SQA (NSW), 4 Hillcrest Place, North Manly, NSW 2100

ASMA/SQA (NSW) Executive Committee 2004/2005*

POSITION	NAME	EMAIL
Chairman	Helen Aitken	helenaitken@ozemail.com.au
Vice Chairman (SQA)	Ted Smillie	consultants@mrose.com.au
Vice Chairman (ASMA)	Sophia Poulos	spoulos@au1.ibm.com
SQA Administration	Jeff Lock	jeff.lock@bigpond.com
Newsletter Editors SQA Newsletter Metrics Matters	Ted Smillie Sophia Poulos	consultants@mrose.com.au spoulos@au1.ibm.com
Committee Members	Michael Adamietz James Leo Redolf Wirawan Daniel Wong Dr Malte Stien Maria Patane	
SQA National Contacts National Chairman Victoria ACT Tasmania WA	Ted Smillie Bruce Hill Shayne Flint Ian Shield Terry Woodings	

About Metrics Matters

<p>Publisher and Copy Editor: Mark Exall Editing: Sophia Poulos</p> <p>Advertisements If you wish to advertise your products or services in Metrics Matters and reach metrics-aware individuals and companies, please contact the Publisher.</p>	<p>Disclaimer The opinions expressed here are those of the individual authors and are not necessarily those of the Publisher, Editor or the Australian Software Metrics Association / Software Quality Association (NSW).</p> <p>Permission is granted to reproduce any item from Metrics Matters on the condition that Metrics Matters and ASMA/SQA (NSW) are fully acknowledged as the source.</p>
---	---

Contributions...

Metrics Matters is seeking contributions from IT Professionals. If you have any form of contribution, please email them to "The Publisher" on info@asma-sqa-nsw.org.au. The document must be in MS Word or Rich Text Format (.rtf). Pictures or advertisements may be in pdf, gif or jpg format.

Last Words...



IT Spending - A Necessary Evil?

By Ted Smillie

The August ASMA/SQA NSW newsletter features a couple of articles attacking the reliability of lines of code and function points for software sizing, and my editorial asks if we are focussing too much on sizing metrics at the expense of quality metrics. Since then I read a paper in the latest issue of the *Journal of Research and Practice in Information Technology* which made me wonder if we are neglecting an even more important measurement – benefits realisation.

The average IT spend is over 4.2% of annual revenue (and rising), so IT managers are under increasing pressure to justify the investment. Easier said than done, since the benefits evaluation process is “a complex tangle of financial, organizational, social, procedural and technical threads, many of which are currently either avoided or dealt with ineffectively.” So say the authors of **IS/IT Investment Evaluation and Benefits Realization Issues in Australia**.

(see http://www.jrpit.acs.org.au/jrpit/JRPIT_Volumes/JRPIT37/JRPIT37.3.235.pdf.)

The findings call to mind a 2001 Compuware Survey of IT professionals in Australia, Singapore & Hong Kong, where most of them said their work was viewed as “a necessary evil.” Sad to say, that view has been around for a while and sometimes with good reason. A landmark 1995 study by Thomas Landauer (*The Trouble With Computers*) identified some of the reasons why business has historically had a poor view of IT and showed there is

justification for the business perception that IT has not delivered on its promise of productivity gains. Landauer identifies the main causes for failure to deliver as inappropriate use of technology, poor understanding of user tasks and inadequate user interface design.

In 1987, Nobel laureate in economics Robert Solow weighed in with “You can see the computer age everywhere but in the productivity statistics.” The absence of correlation between massive IT spending and improved productivity become known as the “productivity paradox”, following a 1987 article by Morgan Stanley's chief economist Steven Roach, titled **America's Technology Dilemma: A Profile of the Information Economy**. Roach noted that the amount of computing power per white-collar worker in the service industry was growing dramatically over the 1970s and 1980s, yet the measured productivity of this sector was flat.

OK, but that was over there and way back then. Where are we here and now? The **IS/IT Investment Evaluation and Benefits Realization Issues in Australia** authors claim that gaining business value from, and justifying current investment in information technology are often identified as the most critical but difficult management issues in Australia, UK and the US. The Australian findings are based on a survey and an in-depth case study of these practices and processes in two large governmental departments with several major IS/IT projects. According to the survey respondents the top three reasons for IS/IT investments were cost/budgets, competitive advantage and process efficiency.

The authors note that the difficulties associated with determining the benefits and costs of IT are deemed to be the

major constraint to investment justification. The difficulty in evaluation centres on the fact that both benefits and costs are difficult to quantify. Therefore, evaluation is often ignored or carried out inefficiently or ineffectively because it is deemed an elusive and complex process. Quoted results from other studies (mainly overseas) include:

- “A considerable amount of IT projects realize no net benefits at all.”
- “45% of the responding organizations did not evaluate whether IS/IT systems were still consistent with business objectives and 59% did not determine whether expected benefits were being achieved.”
- “70% of all IT investments are claimed to give no adequate return on investment.”
- “75% of largescale systems do not function as intended or are not used.”
- “Vague statement of benefits, leading to an uncertain allocation of responsibility for managing their delivery, (are identified) as the number one cause for IS/IT project failure.”

So what do the authors conclude? In general, that many survey respondents and case study participants knew very little about the IS/IT investment evaluation and benefits realization concepts and practices. Most organizations had no formal IS/IT investment evaluation or benefits realization methodologies. The evaluation tended to be informal and quantitative in nature (i.e. based on traditional financially oriented evaluation methods such as return on investment (ROI) and net present value (NPV), or on contractual service level agreements (SLAs).) Most organizations were not

yet mature and even despite the adoption of a formal benefits realization methodology (BRM) some problems still remained. However, the use of a formal methodology in investment evaluation or in benefits realization, would lead to a better IS/IT outsourcing experience for organizations as well as better control over decisions and better attainment of benefits. More broadly, say the authors, it can lead to a higher level of organizational maturity.

The conclusions are persuasive, although I personally have some reservations about the approach taken by the star case study, which involved adopting an external BRM and then setting up a Value Management Office (VMO) “not only to educate the users about the benefits realization methodology but also to minimize user resistance to the methodology.” According to the authors, “This clearly demonstrates the resolve by the organization to implement the methodology as well as to force, if necessary, users to accept the methodology.” Maybe, but to me it starts to smack of the metrics tail wagging the business dog. Where does the VMO sit in relation to organisation’s own program and project management structure? Is it starting to take on the roles and responsibilities normally covered by the organisational change management (OCM) plan as part of project planning?

But that is a quibble. All in all I think the authors make their point well. As to the productivity paradox, that is still controversial but for me the issue was pretty much laid to rest by the work of Brynjolfsson and Hitt, see **The IT Productivity Gap** at <http://www.optimizemag.com/article/showArticle.jhtml?articleId=17700941>.

This article notes:

“Ongoing research, based on data from more than 1,167 large U.S. companies, finds a statistically significant correlation between the intensity of IT used in a company—IT capital per worker—and that company's overall productivity. There's an emerging consensus among economists that IT has been the biggest single factor driving the productivity resurgence, although debate continues about the exact magnitude of its contribution.”

“The critical question facing IT managers today is not *"Does IT pay off?"* but rather, *"How can we best use computers?"* Even when their IT intensity is identical, some companies have only a fraction of the productivity of their competitors.

“We do find evidence of a substantial relationship between computers and productivity growth, but closer examination reveals that the biggest benefits accrue to companies that adopt an identifiable cluster of business practices we call the digital organization. Relatively few of these practices are directly related to the implementation of the technology itself. Most involve changes in the organization of information work, including decision-making rights, incentive systems, hiring, and training.

Not all companies, even among aggressive users of IT, have successfully implemented these practices. But companies that are unusually productive have overcome the adjustment costs associated with organizational innovation. They've adopted a new cluster of information work practices.”

Ted Smillie is Director, Quality Assurance Services at Montrose Computer Services. He is current National Chairman, Software Quality Association (SQA) and former ASMA/SQA NSW Chairman. His specialties are Software Process Improvement, Software Project Management, Software Quality Assurance & Testing and Software Risk Management.

✂