COSMIC-FFP and the economic case for software metrics

Charles Symons

Joint Project Leader
The Common Software Measurement International Consortium

UKSMA Conference October 2006

Agenda

- The current state of software metrics
- The Critical Success Factors for a software metrics programme
- Why a credible software sizing method such as COSMIC-FFP is vital for success
- What could be the economic benefits for a successful software metrics programme?

Software metrics are STILL not mainstream

"You cannot manage unless you measure"

RUBBISH – PEOPLE DO 'MANAGE'

The statement should be: "Managing software projects without metrics increases RISK"

Meta Group 2002:

- 89% of organizations collect no performance measurements at all on their projects, apart from financial information"

Standish Group CHAOS report 2003:

- 34% of projects are 'successful'
- 51% are 'challenged'
- 15% are 'failures' (representing 1/3 of the cost)

And customers pay heavily for the software industry's ignorance of quantitative methods

Example

The industry promotes methods like 'RAD' and 'XP' to deliver functionality earlier than traditional methods, but ….

Q's: Which methods have the

- least overall development costs?
- least overall development time?
- least overall life-cycle costs?
- best delivered quality?
- how do the answers vary with software size?
- etc

And many software metrics programs have limited effectiveness

Howard Rubin

"The average life of a software metrics program is three years"

Hetz 1993

Most of the data collected is that which is easy to collect (% orgs. collecting):

> 50%: no. of defects after release, no. of changes, customer satisfaction, etc

<20%: documentation size, re-used SLOC

<10%: function points

Some Critical Success Factors for a software metrics program

- Metrics must be credible to those measured and to management
- The metrics activity must be complementary to the processes measured
- The metrics effort must be acceptable
- The organization and its processes must be reasonably stable over enough time to gather sufficient metrics for worthwhile analysis
- The measurements must be used to help achieve the organization's goals

Measurement of a software size is the foundation for any metrics programme

Uses:

- **Project planning**
- **Estimation of effort, etc**
- **Project scope and contract control**
- **Performance measurement (productivity, speed of delivery, quality) and benchmarking**

… for development and maintenance activities

So let's examine how three common software size measures can support the metrics programme CSF's

CSF ↓ / Size metric →	SLOC	IFPUG FPA	COSMIC -FFP
Credibility			
Complementary			
Effort OK			
Process stability	(Size measurement method has no influence)		
Results used			

COSMIC-FFP is the first size measure that passes all the credibility tests

SLOC	IFPUG FPA	COSMIC -FFP
• Can be measured precisely, but rules vary • Size is technology-dependent • Language conversion factors questionable • Used only for real-time & embedded software	• Pragmatic sizing rules based on some IBM software >25 years ago • Limited size scale (non-linear, ordinal); lacks credibility for large, complex software • Large existing base of performance measurements – but only for business application software	• Designed by an international team based on sound software engineering principles • Designed to measure business application & real-time software in multi-layer, multi-tier architectures, • A ratio size scale • Growing user & measurement base

COSMIC-FFP is the only method that complements modern software engineering practice

SLOC	IFPUG FPA	COSMIC -FFP
• Can only be measured accurately on project completion • Can only be 'guestimated' from requirements – limits it use for estimating • Less-skilled programmers (& cheats!) will produce larger SLOC sizes	• Underlying concepts are irrelevant to modern software engineering practice • Measurement is a separate process, hence difficult for a project team to understand and accept the results	• Underlying concepts complement modern SE approaches such as UML but are independent of any one method • Measurement can be embedded in project processes, minimizing data collection cost • Measurement using COSMIC-FFP assists quality control of requirements

The effort for COSMIC-FFP size measurement is acceptable

SLOC	IFPUG FPA	COSMIC -FFP
• Measurement of completed software can be automated	• Manual measurement but acceptable effort	• Manual measurement but acceptable effort • Can be integrated in project processes • Measurement may be at least partly automatable if requirements are held in a CASE tool

Conclusion

The COSMIC-FFP functional size measurement method could make the difference between a credible software metrics programme and one that will fail and be abandoned

So what might be the benefits to the UK software industry* if it used metrics (where relevant) to improve

1. Process performance

2. Estimating accuracy?

(The use of metrics to improve project control is assumed to be included in 1 and 2)

*** The UK spends £20 Billion pa on software (‘‘Survey-based measures of software investment in the UK’, Office of National Statistics, Feb 2006)**

1. Process performance: some conservative assumptions

Assume

a. Only half the software industry can benefit from use of metrics
b. Industry doubles productivity over 10 years
c. Software Process Improvement (SPI) requires an additional investment of 5% pa
d. The contribution of metrics to SPI is proportional to its share of the costs, say 5%
e. So the share of the benefits attributable to metrics is also 5%

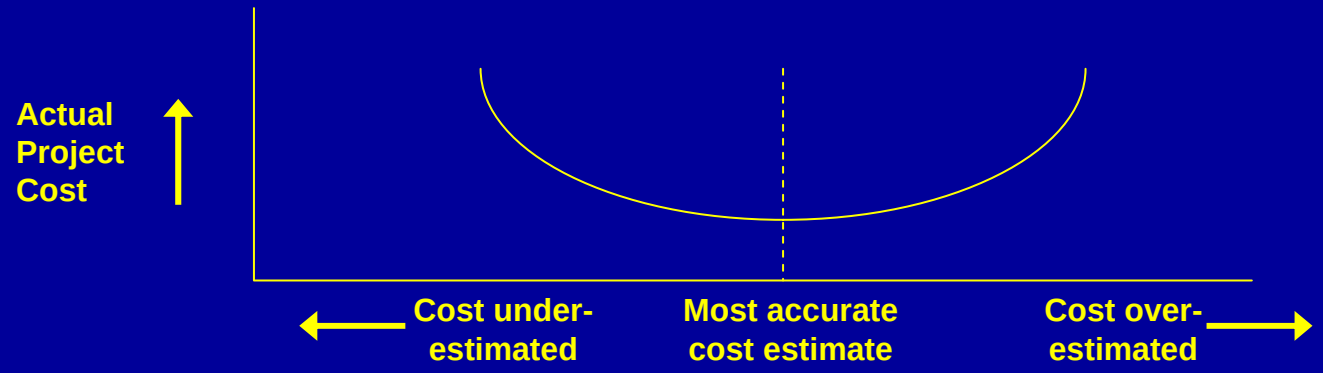
Then the net benefit of SPI averages ~£4.4 Billion pa over the 10 years

And the contribution to this net benefit of metrics is ~£220M pa.

2. Estimating accuracy: some conservative assumptions (1)

Assume

- **On average estimating accuracy could be improved by 10% with proper use of metrics**
- **The most accurate estimate leads to the lowest project cost***



- **The Standish CHAOS report findings on project success/failure rates apply to the UK software industry**

* 'The elusive silver lining: how we fail to learn from software development failures', Abdel-Hamid, Madnick, Sloan Management Review, Fall 1990

2. Estimating accuracy: some conservative assumptions (2)

- Projects initially over-estimated that go ahead (% unknown) cost 10% more than necessary
- Projects initially over-estimated that don't go ahead (% unknown); their costs and benefits, are 'lost'
- Projects initially under-estimated; some will be stopped. These are the 15% of all projects (33% of costs). Assume better estimating on this group alone would save 3% of their costs, i.e. 1% of software investment
- Projects initially under-estimated that run to completion; these cost more than necessary – the 'challenged' group, 50% of all projects. Assume better estimating could save 1% of their costs, i.e. 0.5% of software investment

The bottom line: better estimating could lead to >1.5% cost savings – worth £150M pa to the UK software industry

Conclusion

With very conservative assumptions, proper use of software metrics could directly benefit the UK software industry by at least £400M pa …

(or > 2% of the software investment of any organization where metrics are relevant)

… and indirectly assist in obtaining net benefits of ~£4.5B pa

Am I alone in thinking this way? (No)

Example: Two of seven Principles of SMART acquisition *:

- **Principle 1. Adopt a whole-life approach, typified by applying through-life costing techniques**
- **Principle 5. Establish effective trade-offs between system performance, through-life costs and time**

- **And 'loose approximations suggest 15% of total procurement spend should be for de-risking'**

*** UK House of Commons Defence Committee HC 572 Session 2003/4, 28 July 2004 into MOD acquisition (including the acquisition of software-intensive systems)**

Two case histories remind us of the scale of the problem

A UK Govt Department recently received two bids in response to one ITT – for £0.25M and for £2M

A City of London financial institution recently cancelled a software project on which £40M had been spent

(Early in the project it was estimated that the size would be ~ 100,000 IFPUG FP's, a 'mission impossible' project. Management ignored the advice)

And another case reminds us that the problems can be solved

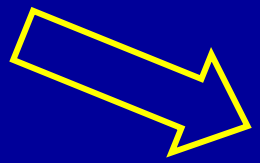
One of the world's largest telecoms companies uses the COSMIC-FFP method for performance measurement and estimating, and reports:

" ……. The results for improving estimation accuracy by means of COSMIC-FFP based size measurement are very encouraging, especially the larger an estimated item, the better the result that is achieved."

In summary …

A successful software metrics programme needs credible size measures that complement project processes

(e.g. COSMIC-FFP)



Proper use of metrics is key to unleashing huge financial benefits for software producers

Thank you for your attention