MEASURES FOR EXCELLENCE

Analysis of On-Board Spacecraft Software Development

J.W.E Greene

QUANTITATIVE SOFTWARE MANAGEMENT LTD

7 rue Fenoux
Paris 75015
Tel: 33-140-431210
Fax: 33-148-286249
qsm.europe@ pobox.com

93 Blythe Road,
London W14 OHP
Tel: 44-171-603-9009
Fax: 44-171-602-6008
www.qsm.com

1 Introduction

The percentage of development costs attributed to software is growing rapidly, particularly in high technology products and engineering systems. In addition the risks in development are increasingly associated with software. Purchasing organisations are faced with evaluating development proposals with an ever larger and sensitive software content.

Many procurement managers have little or no background in software. Frequently these managers find it difficult to get valid numbers to help make informed decisions on the procurement of software. Their perception is that there is very limited, or no data, on software projects related to their needs.

Our experience shows this situation is remedied by collecting a modest amount of data that is readily available from completed and planned software developments. Concern is often expressed of the difficulty of collecting such data. The data described in this paper is always obtainable from recently completed and planned software developments, in this instance on-board spacecraft developments.

On-board spacecraft software is an example of a specific type of software application development. In contrast with many other types of software application, such as radar or command and control systems, there are still relatively few on-board spacecraft software systems. The case study described in this paper confirms the availability of basic data.

Collecting the basic data and the analysis described in this paper took approximately one month. Data collection was undertaken with the Agency personnel familiar with each on-board development. Contractors provided data on development plans giving estimates for proposed projects. Experience indicates this contractor data is forthcoming, particularly where the data is made mandatory to support development proposals. (Ref. 3)

2 Measurement Objectives

Before describing the software development data and the on-board measurement results, it is useful to set out the objectives sought from getting these measures. The objectives are to:-

- evaluate the data from completed developments against an independent database of real-time software projects to determine their characteristics
- measure the process productivity of each development using QSM's engineering model of development team behaviour
- compare the resulting process productivity values against the expected measures for real-time applications
- similarly evaluate the proposed plans for new software developments and compare these against the history from the completed projects

- re-estimate the plans for the new developments in a consistent manner and compare against completed projects.
- demonstrate the practicality of collecting data and how the measures assist procurement managers by carrying out all of the above.

3 The Software Development Data

Our method of analysis of software projects requires minimal data on completed, in-progress and planned projects namely :-

- software development time in months
- software development effort in man months
- software size in effective lines of code.

Each of these quantities is readily available. The QSM technique operates at a "macroscopic", that is "big picture" level, concerned only to measure and analyse software development team behaviour.

4 Project Data: Completed and Planned Developments

Data from completed projects was available from the Agency staff associated with each development. Proposed projects are analysed using the data from contractor development proposals.

Development time is in elapsed months and development effort is in person months. The size in effective lines of code (ELOC) is given for each project. These three values constitute the minimum data required to perform the analysis described.

The data set out below was collected for analysis from completed projects (CP) and proposed projects (PP).

4.1 Completed Projects

Completed Project	Development Time	Effort	Software Size(ELOC)
CP 1	20	27	4000
CP 2	18	21	4500
CP 3	24	24	6000
CP 4	50	200	25000
CP 5	26	91	26000
CP 6	42	300	30000
CP 7	40	560	55730
CP 8	54	2400	200000

Copyright QSM 93 Blythe Road, Brook Green, London W14 OHP
Tel: 44-171-603-9009 Fax:44-171-602-6008 (SPACE96)

The first three completed projects, CP1, CP2 and CP3 are very small research developments undertaken by University personnel. Typically these involve only one or two people.

The remaining five completed projects have been developed by professional development teams from contractors and, in one case, the Agency itself.

4.2 Proposed Projects

The data for the proposed projects are estimates. Each development plan states the proposed time and effort. In addition the supplier makes a size range estimate for the software giving the mean size.

Proposed Project	Development Time	Development Effort	Software Size (ELOC)
PP1	24	240	8000
PP2	30	98	8000
PP3	36	185	15000
PP4	30	300	22000
PP5	36	245	22000

5 The QSM Reference Database: Fundamental Software Trendlines

The analysis of software project team behaviour and performance uses the database we have assembled from developments in Europe, the USA and Japan. Currently we have examined data from more that 8000 projects of all application types, validated the data and assembled over 4000 recent projects in detail, to analyse with high confidence (Ref. 1).

To analyse the embedded real-time systems from the Agency we have used the trend lines derived form similar systems in our database. These amount to approximately 300 real-time systems., The "trend lines" shown in the figures 1 and 2 are produced to provide baselines for comparison by determining the least squares fit through these data points and the standard deviations. Note that the trend line graphs in both figures have logarithmic scales. The trend lines show the basic behaviour pattern of the development time and effort as a function of system size

These baselines are used with the basic data to position projects and determine their specific development time and effort characteristics. In this case the results give significant insights into development team behaviour by analysing the completed and proposed on-board spacecraft software development projects.

Copyright QSM, 7 Rue Fenoux 75015 Paris Tel : 33-140-431210 Fax : 33-148-286249 (SPACE96)

6 Completed Projects : Development Time and Effort Trend Line Analysis

For each completed project we position the development time and effort against the trend lines for the corresponding software size. The results are shown in Figure 1.

All the projects are positioned at or above the mean time and effort expected for their size. The characteristic behaviour in these completed projects is to extend time beyond the mean. Bearing in mind the logarithmic scales, this time extension is large for the majority of the developments. The corresponding effort is found to be around the mean value for almost all projects. All are within one standard deviation.

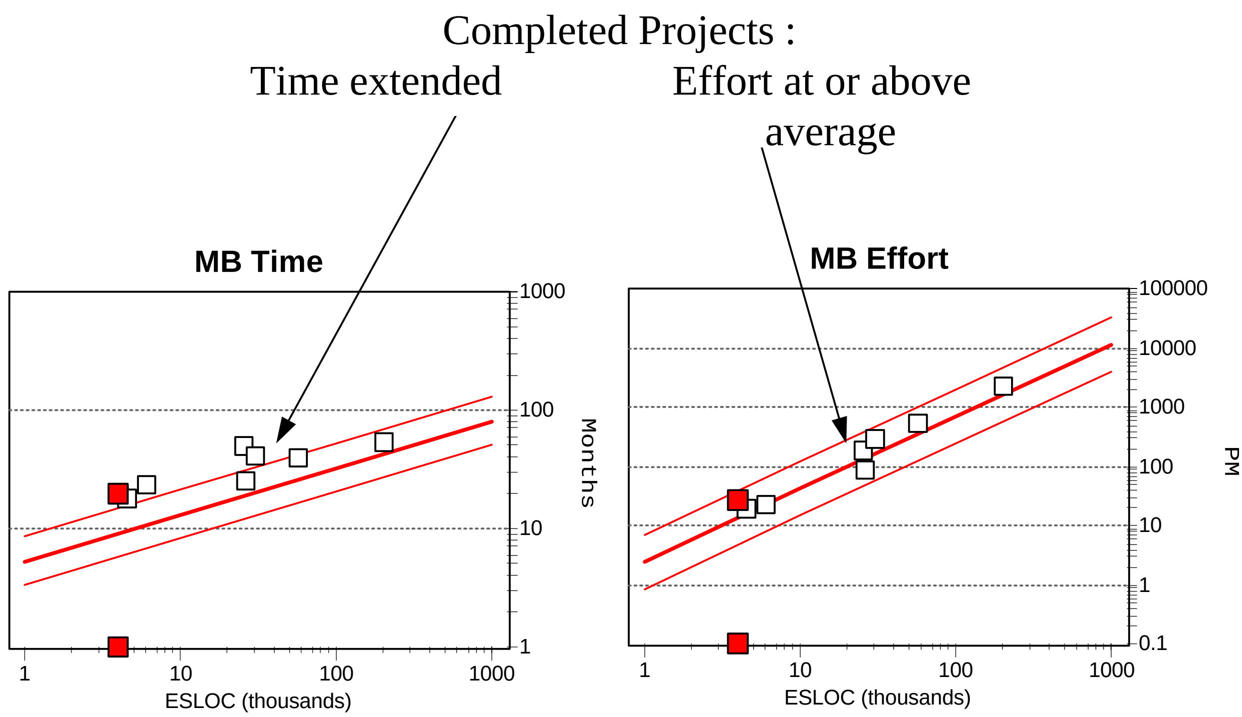

FIGURE 1 COMPLETED PROJECTS: DEVELOPMENT TIME AND EFFORT VERSUS QSM REAL-TIME TREND LINES

As shown in Figure 1, completed on-board software developments have very characteristic development times and effort compared with our independent database of real-time projects: -

- development time is extended
- the corresponding effort is around average

7. Planned Projects : Development Time and Effort Analysis

For each planned project we position the proposed development time and effort against the corresponding software size using the same trend lines of the development time and effort extracted from our real-time database projects. The results are shown in Figure 2.

All the proposed projects expect to take substantially longer than the mean time for their size. Indeed all of the projects are planned to take one standard deviation longer than the mean time. With the exponential scale on the vertical axis, this represents a very considerable time extension compared with the mean value for real-time systems.

Figure 2 shows the proposed development effort for the planned projects. All the projects expect to take substantially more effort at or above one standard deviation. This amount of effort contradicts the history from completed developments. We see in Figure 1 that effort in completed projects is typically within one standard deviation.

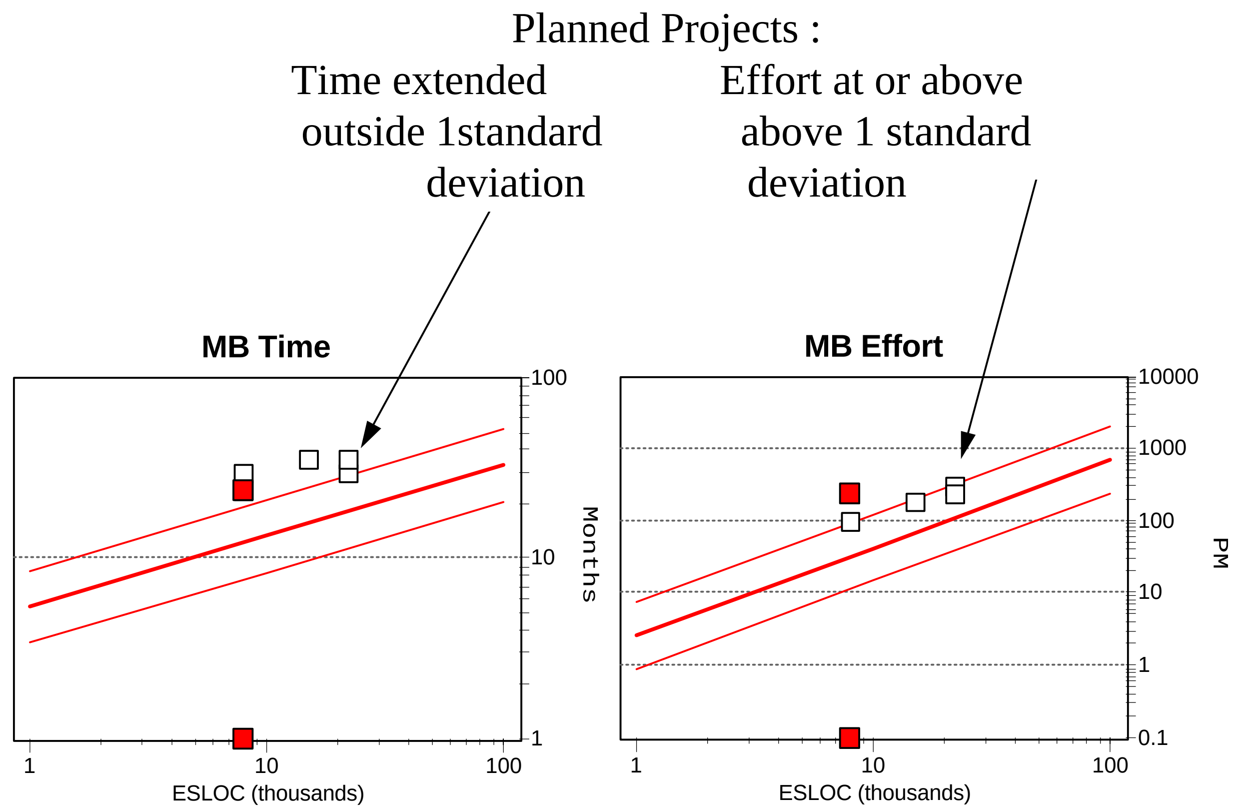

FIGURE 2 COMPARISON OF PROPOSED MAIN BUILD TIME AND EFFORT VERSUS THE QSM REAL-TIME TREND LINES.

In the following sections we analyse the basic data from the completed and the proposed projects in more detail using QSM's software engineering equation that describes the software development team behaviour.

8. QSM Software Equation - Process Productivity Measures

QSM's research and analysis (2) provides an equation that links the functionality of the software created to the time and resource expenditure required producing it. We call this the software equation. All the equation terms that determine effort and cost are exponential.

Copyright QSM, 7 Rue Fenoux 75015 Paris Tel : 33-140-431210 Fax : 33-148-286249 (SPACE96)

Conceptually the software production equation can be thought of in the simplified form such that:

Product = Process Productivity * Effort * Time
Where:

- **the product** is the software function created or modified. Generally this is expressed in delivered executable source lines of code (ELOC).
- **process productivity** is a proportionality parameter which reflects software complexity as well as the development environment factors unique to the developer.
- **effort** is the cost in total man months of work by all staff categories for the software development phase
- **time** is the elapsed calendar months for the software development.

For more details on the QSM software equation applied in a procurement organisation, including an analysis of ground based software developments, a comprehensive exposition has been made by the French Space Agency, CNES (Ref. 2). This analysis examined data from over 20 software projects and measured their process productivity using the QSM approach. The analysis sets out re-estimates of proposed developments consistent with observed measures, including an on-board spacecraft development.

<u>8.1 QSM Process Productivity Index Measures : Completed and Proposed Developments</u>

We express process productivity using a term derived from the software equation that is called the Productivity Index, (PI). The PI is represented on a straight forward linear scale from 1 to 40 (actual values are discreet numbers.) All factors in the development environment, which influence the software development team, including application complexity, are reflected in the PI measure.

World wide, measures have been made on more than 8000 projects. From the measures there are expected average PI values for each type of application. Below are set out expected average values for a selection of the major software application types that we analyse.

Software Application Type	Expected Average Productivity Index
Microcode	6
Real-time Systems	7
Avionic Systems	7
Radar Systems	7
Command & Control Systems	8
Telecom Systems	11
Scientific/Engineering Systems	12
Business Systems	17

Compared to the expected average PI value for each application type:

Copyright QSM 93 Blythe Road, Brook Green, London W14 OHP
Tel: 44-171-603-9009 Fax:44-171-602-6008 (SPACE96)

- lower values measure low process productivity
- higher values measure high process productivity .

The QSM software engineering equation is used to calculate the PI achieved in each project first by using the development time, development effort and the size of each of the completed projects,. Figure 3 in the left hand histogram shows the results obtained by plotting a simple histogram of the PI values found in completed projects. Here the Productivity Index values cluster to the left of the expected industry average. These indicate lower than average process productivity in the completed projects. We also show the current expected industry average value for real-time development (PI=7) and the +/- one standard deviation limits.

In the same way the size, development time and development effort are used from the proposed development plans to calculate the implied Productivity Index for each planned project. The right hand part of Figure 3 shows the results from our analysis expressed as a simple histogram.

In the proposed projects three indicate the contractor expects to achieve the lowest value on the scale, 1. Two other projects indicate they expect to achieve a Productivity Index of 3.

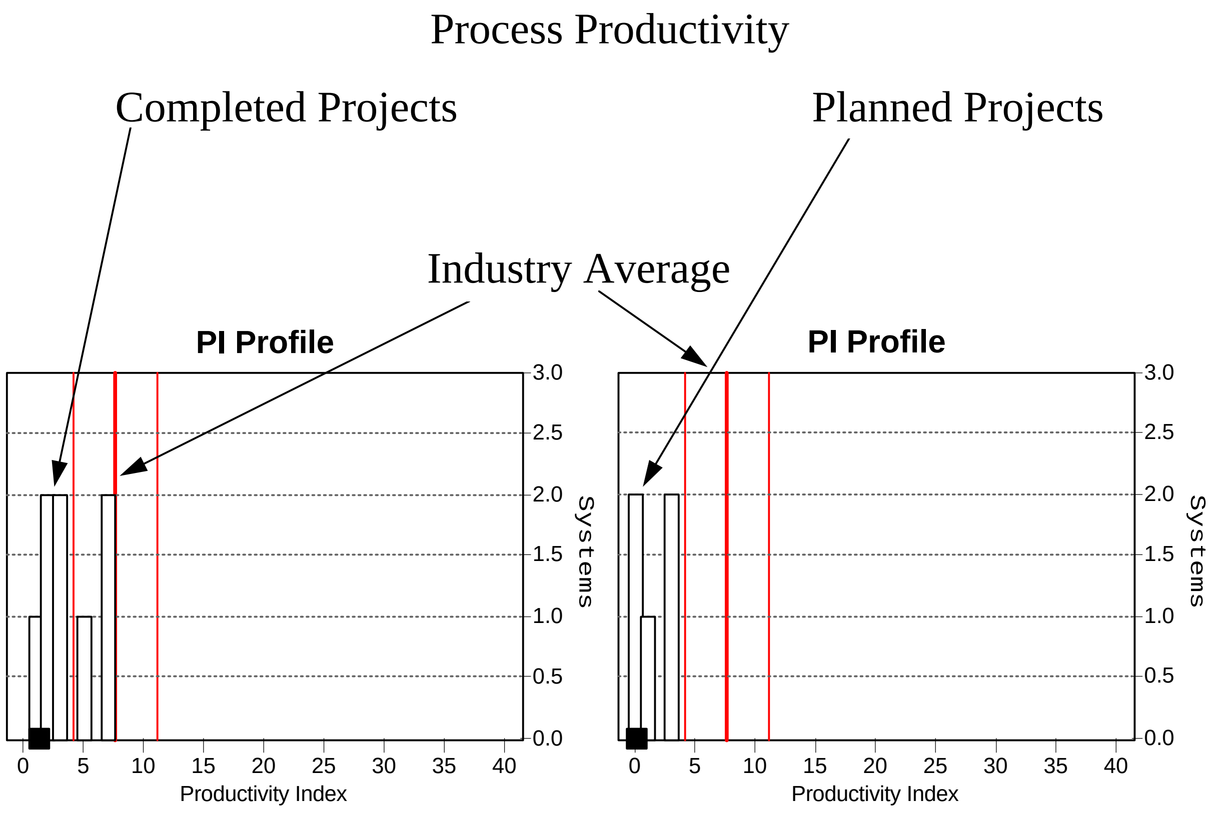

FIGURE 3 :HISTOGRAMS OF PROCESS PRODUCTIVITY INDICES MEASURED IN COMPLETED AND PLANNED PROJECTS

Copyright QSM, 7 Rue Fenoux 75015 Paris Tel : 33-140-431210 Fax : 33-148-286249 (SPACE96)

When compared with the completed projects Productivity Indices, we see that historically only one project has been found with the lowest PI value of 1. The low process Productivity Indices measured in the proposed projects indicates extremely unproductive development environments where contractors expect to use high effort in relation to the size of software developed.

From this analysis of the completed and proposed software projects PI's, we conclude the contractors appear to be unduly pessimistic with regard to their productivity. The proposed plans assume very high effort in relation to the size of software to be developed and the evidence from completed projects.

To arrive at plans which are consistent with contractors at least maintaining their development process productivity, each proposed project is re-estimated. This gives development effort consistent with the completed projects, given the contractors achieve the average process Productivity Index of 4 measured in the completed projects.

9. Re-estimated Projects

Using the average value of the process Productivity Index measured in completed projects, we re-estimate the development time and effort for the new development proposals. Our estimates assume a value of four in each development. The results are shown in Table 4 based upon the mean size of each new development. Note that where the mean size is the same for the first and the last two projects then we re-estimate consistently to give the same time and effort.

Table 4 QSM Re-estimate Development Time and Effort : Process Productivity = 4

Re-estimated Project	Size ELOC	Re-estimated Time	Effort	Proposed Time	Effort	Excess Effort
PP 1	8000	19	40	24	240	+200
PP 2	8000	19	40	30`	98	+48
PP 3	15000	25	90	36	185	+95
PP 4	22000	29	200	30	300	+100
PP 5	22000	29	200	36	245	+45

9.2. Revised Estimates Comparison

By taking the results for time and effort from the revised estimates we position these values against the real-time trend lines. This provides a check on the development re-estimate values when compared with the completed projects. Figures 3 show the results of the comparison.

Examining Figure 3 shows the re-estimated time for each proposed development is longer than the mean time for it's size. Now however, we see that in every case the corresponding effort is around the average expected for its size and within one

Copyright QSM 93 Blythe Road, Brook Green, London W14 OHP
Tel: 44-171-603-9009 Fax:44-171-602-6008 (SPACE96)

standard deviation of the mean value. By basing our new estimates on the insights provided from completed projects we arrive at more consistent results for developments and hence have more confidence in both the time and the effort being proposed.

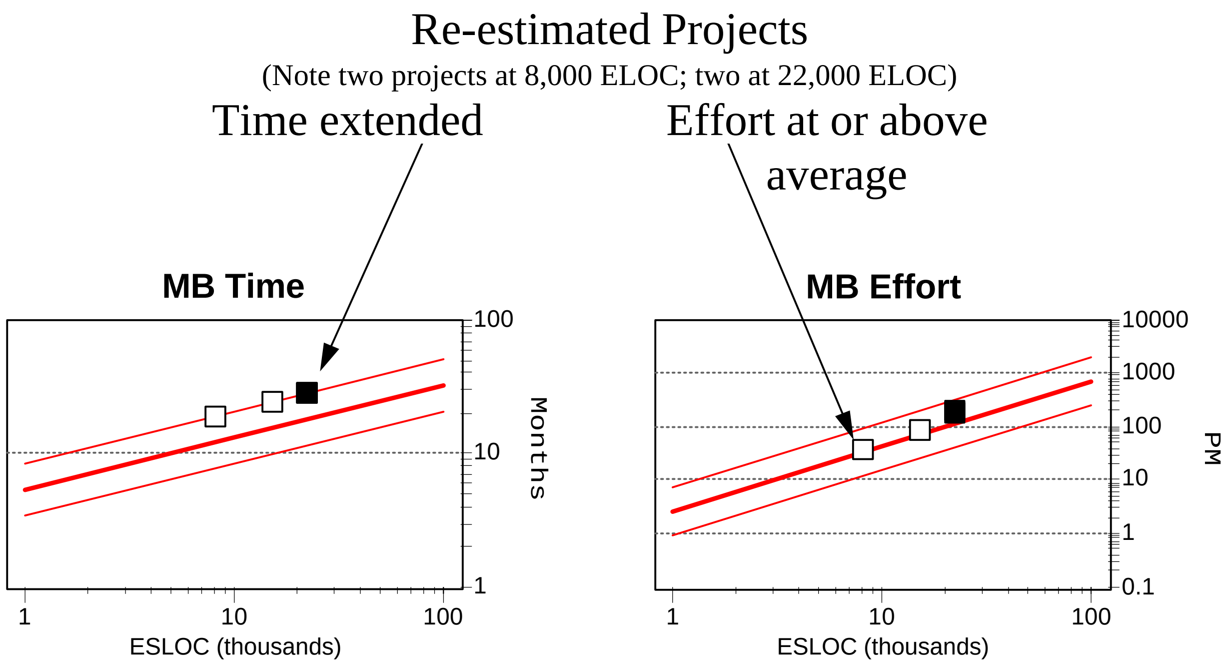

FIGURE 3 : QSM ESTIMATED MAIN BUILD TIME AND EFFORT VERSUS THE QSM REAL TIME DATABASE TREND LINES

10. Observations

The analysis of the completed projects reveals significant insights into the behaviour of on-board spacecraft software development. In particular for the completed projects:

- development time is extended
- the corresponding development effort is within one standard deviation

This evidence is also consistent with similar analyses we have made on other complex real time systems involving innovative hardware. Design trade-off's are frequently made between software and hardware. Also the specifications of complex interfaces are only finalised during development. These factors act as "natural" constraints, which limit the number of software development staff it is sensible to employ and cause time to be extended.

New developments propose to extend the time for the size of software to be produced, which is broadly consistent with the completed projects. However the effort proposed is significantly more than what might reasonably be expected from the completed development behaviour.

The effort proposed indicates that developers are assuming their development process productivity is reducing in these developments. We would expect productivity to be improving, rather than reducing, given the experience with this type of application as well as improvements in the development environments.

Use of the QSM software equation enables measures of process productivity to be made in the completed projects. These productivity measures are consistent with what we would expect in embedded real-time systems. Revised development effort estimates are made using these measures for the developments, which are consistent with the completed projects.

11. Conclusions

The main conclusion is that it is practical and beneficial to collect and analyse basic software development data from application developments which are relatively rare. In the majority of application types there are many more projects capable of similar analysis. With the minimal data described in this paper, procurement managers are given a clear understanding of the characteristics of their software developments.

A database is built to provide a reference source for each application type of interest to the procurement organisation. This permits unique development characteristics to be identified. Armed with these insights, independent baseline estimates are made to give procurement managers an early understanding of the likely timescales and costs of new software developments.

A suitably structured Invitation to Tender (ITT) (Ref. 3) permits the informed analysis of development proposals. In addition the ITT requests basic data on similar completed projects to confirm the contractors capability. New proposals are compared against the completed developments to ensure they are consistent. Analysis reveals those contractors who are most productive and who can demonstrate their capability from results in completed projects.

These insights are applied to assess contractors and confirm their proposals offer value for money and ensure the development proposals are consistent with completed projects. Significant benefits result including realistic estimates of development time and cost.

In the case of the five proposed on-board spacecraft software developments analysed in this study, the savings from realistic development effort estimates are of the order of 40 man-years or approximately $8 million.

Ref 1. Putnam L.H. and Myers W. “Measures For Excellence - Reliable Software, On Time and Within Budget” Prentice Hall ISBN 0-13-567694

Ref 2. Ricard P.A. "Estimation des Parametres de la Production Logicielle par Model Economique; Resultats Preliminaries Obtenus an CNES" Centre National D'Etudes Spatiales, 31400 Toulouse, France 1983

Ref 3. Kempff G.K. KPN Logistics Handboek Software Control Management :Royal Dutch PTT Supplier Tender Evaluation

Copyright QSM 93 Blythe Road, Brook Green, London W14 OHP
Tel: 44-171-603-9009 Fax:44-171-602-6008 (SPACE96)