MEASURES FOR EXCELLENCE

THE (ALMOST) PERFECT SOFTWARE PROJECT

USING

THE SEI CORE MEASURES

QUANTITATIVE SOFTWARE MANAGEMENT LTD

7 Rue Fenoux		93 Blythe Road
75015 Paris	Internet: qsm.europe@pobox.com	London W14 0HP
Tel: 33-140-431210	CompuServe: 100113,3364	Tel: 44-171-603-9009
Fax: 33-148-286249	www.qsm.com	Fax : 44-171-602-6008

THE (ALMOST) PERFECT SOFTWARE PROJECT USING THE SEI CORE MEASURES

Introduction: The SEI Core Measures

Is it possible to successfully plan and manage software development with minimal data? The Carnegie Mellon Software Engineering Institute (SEI) recommends that four core measures be made on software developments, namely software size, time, effort and defects. [1] So the interesting question is can software development be done with these core measures?

The only way to prove the practicality and the benefits is to use the core measures and show the results. The background to the development set out here involves the purchasing department of a Telecommunications Operator (Telecom) who insists that all development proposals be quantified using the core measures. [2]

First the Telecom checks the proposal plan is realistic. This plan data allows a quantified baseline contract to be agreed. The supplier is then contractually required to provide progress data at least every month. The progress data is used to evaluate and report progress. The goal is to ensure that delivery of the full function is on time, within budget, and the software is delivered with high reliability.

Naturally suppliers are motivated to get the Telecom's business and hence to supply the data on the plans and progress. The core data allows the Telecom to assess each supplier proposal quantitatively. These measures complement the CMM Maturity Levels [3] that are also used to by the Telecom to assess the qualitative factors in the supplier's development process.

In the development described here it was the first time the supplier had been requested to provide the plan data using the core measures. In particular the requirement to estimate the expected size range of software was completely new.

It is worth noting a recent report dealing with software purchasing to understand why purchasers of software development should be motivated to use the core measures. (Ref. 4.) This is a critical evaluation of the software purchasing competence of the US Federal Aviation Authority (FAA). The report sets out how the FAA is exposed commercially without getting and using core data.

The Perfect Project Plan Data

Before contract award the supplier was required to estimate the size range of the software to be developed. The size range is expressed in logical input statements (i.e. what is to be written by the team) and estimated as the minimum, most likely and maximum values. [5] This takes in to account the uncertainties in the requirement specification by estimating the size range of each software module. The supplier did this based on the 18 modules identified for development. The result is shown in Figure 1.

The proposed development-staffing plan for 13 months was provided and is shown in Figure 2.

SIZE ESTIMATE IN LOGICAL INPUT STATEMENTS (LIS)

Complete for all Modules		All Modified and New Modules ESTIMATED SIZE RANGE MODIFIED + NEW LIS		
MODULE ID		Least	Most Likely	Most
1	MOD 1	2000	2500	4000
2	MOD 2	2000	3000	6000
3	MOD 3	2800	3000	4000
4	MOD 4	1000	1200	1500
5	MOD 5	2000	3000	4000
6	MOD 6	2000	2000	3000
7	MOD 7	800	1000	1200
8	MOD 8	2000	3000	6000
9	MOD 9	1500	2000	2500
10	MOD 10	1000	2000	3000
11	MOD 11	500	1000	1500
12	MOD 12	2500	3000	4000
13	MOD 13	500	1000	1500
14	MOD 14	300	500	800
15	MOD 15	500	1000	1500
16	MOD 16	500	1000	1500
17	MOD 17	500	1000	1500
18	MOD 18	500	1000	2000
TOTAL		22900	32200	49500

Figure 1: The module size range estimate data

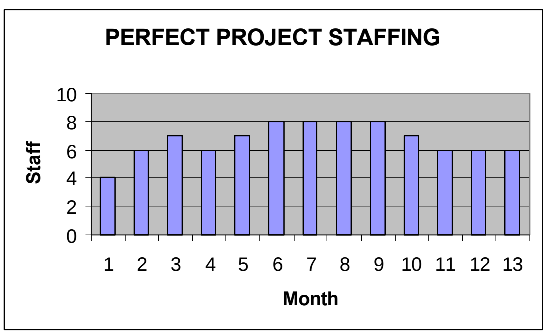


Figure 2: The software main build (MB) time and effort.

Using the Core Measures to Evaluate the Plan

The core plan data of size, time and effort allows comparison against industry reference measures available for different application types. [6] In this case the comparison is made against Telecom developments and the plan can be confirmed as realistic and within the bounds of known industry values.

In the project the basic data are the expected size at 32,000 LIS, development time of 13 months and the total effort planned at 87 person months. Figure 3 shows the main build (MB) plan (Black Square) compared against the Telecom industry trend lines derived from a database of similar developments.

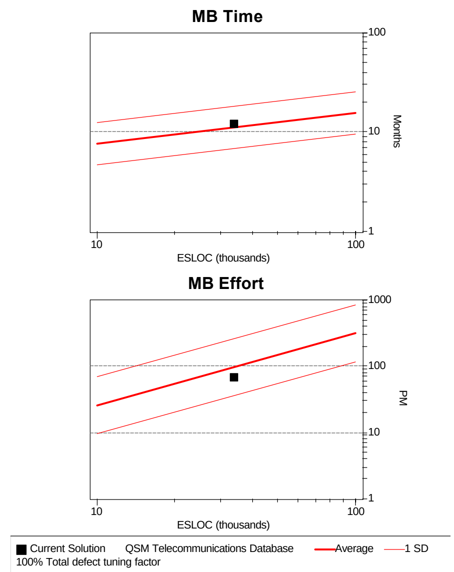


Figure 3: Comparing the Planned Size, Time and Effort against Industry Reference Measures

The core planning data shown above is also used to calculate the process productivity of the development team assumed by the supplier. In this case the process productivity value is determined at 12.5. This is consistent with the expected industry average Telecom development value of around 12. [6]

So the “Health Check” on the plan shows it is in line with expected industry values. The plan now forms the baseline to track and report development progress.

Contractual Progress Data

Mandatory contract progress data is returned every 2 weeks and is used to track progress and identify if there is any risk of slippage. The progress data is used to perform variance analysis (a form of statistical control) against the baseline plan.

The progress data consists of:

- Staffing- how many people allocated to the project
- Key milestones passed- for instance program design complete, all code complete
- Program module status- is it in design, code, unit test, integration or validation?
- Program module size when the code is complete
- Total code currently under configuration control
- Software defects broken down in to critical, sever, moderate and cosmetic
- The number of planned and completed integration and validation tests.

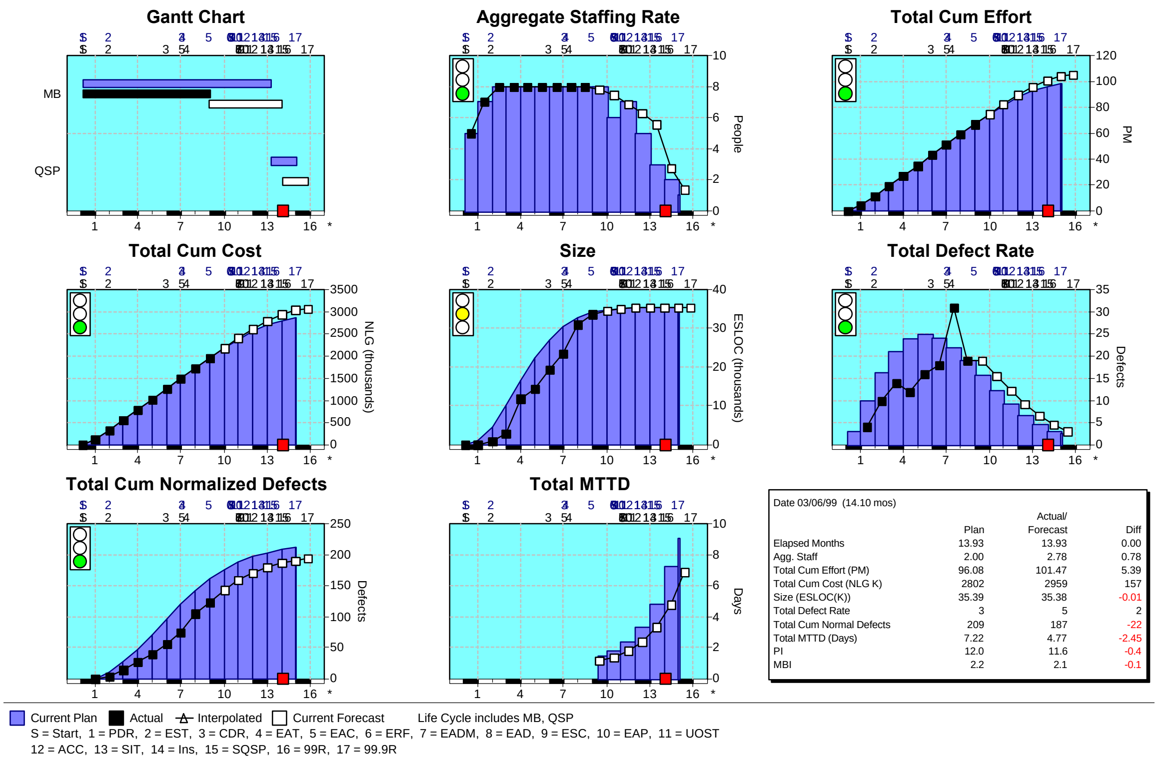
This progress data is essential for the management of software development. Without this basic data the development is out of control.

Tracking and Reporting Progress and Forecasting Completion

Each reporting period the progress data is used to determine the position in the project against the baseline plan. Advanced statistical control techniques use the data to determine if there is significant variance against the plan. Risk is reported using stoplights. If significant variance is found then weighting algorithms enable the new completion date to be forecast as well as forecasting the outstanding data to complete. This can include code production, defects and tests. [6]

Figure 4 shows the situation in the development project after 9 months. The black squares represent actual reported data. White (open) squares and lines are the forecasts determined from the progress data. Stoplights are used to highlight significant variance. An Orange light is shown for size, here the total code production has only just reached the expected size.

Defect behavior is of particular interest. In Figure 4 the total defect rate is following the expected theoretical curve with occasional excursions. These rate variations are smoothed out in the accompanying cumulative curve. The corresponding mean time to defect (MTTD) is indicating high reliability at delivery.



Date 03/06/99 (14.10 mos)

	Plan	Actual/ Forecast	Diff
Elapsed Months	13.93	13.93	0.00
Agg. Staff	2.00	2.78	0.78
Total Cum Effort (PM)	96.08	101.47	5.39
Total Cum Cost (NLG K)	2802	2959	157
Size (ESLOC(K))	35.39	35.38	-0.01
Total Defect Rate	3	5	2
Total Cum Normal Defects	209	187	-22
Total MTTD (Days)	7.22	4.77	-2.45
PI	12.0	11.6	-0.4
MBI	2.2	2.1	-0.1

Figure 4: Variance (Risk) Analysis and Forecast to complete

Keeping the Audit Trail

A criticism of the FAA [4] is that an incomplete record is kept of the changing plans and forecasts in projects. The system used to track the Telecom development allows all plans and forecasts to be logged. At the end of the project there is a complete history in terms of plans, progress at a given date and the corresponding forecast.

This capability is shown in Figure 5 where the solid (blue-lighter) lines are plans while the solid (black) parts of lines are actual progress data with the outstanding forecast shown as white. Each entry represents a plan or forecast logged at a specific date.

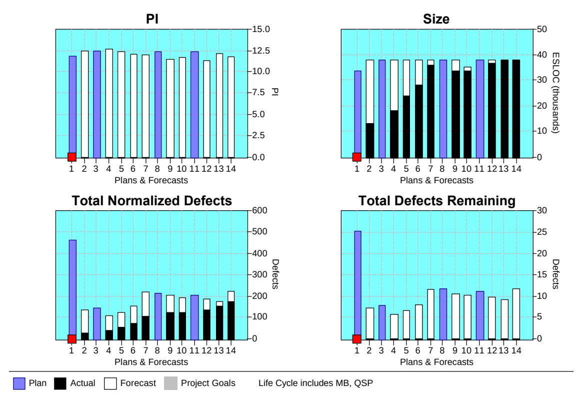


Figure 5: Logged Plans, Actual Data and Forecasts

Controlling Requirement Changes

At month 10 in the development a change request was raised. Using the size baseline (this was being confirmed by the actual code produced) it is practical to evaluate the impact of such a request. To do this the total size is increased by the size estimated for the change request. Using the new size a forecast is made of the new delivery date as well as the additional staffing needed.

Here the results showed that an unacceptable delay would result so it was decided to postpone the change to the next release.

Final Tests and Acceptance

Once the code is complete then the main activity in development is to execute the integration and validation tests. These tests detect the remaining software defects and characterise almost all software projects. Figure 6 shows the planned tests and weekly progress towards the end of the project.

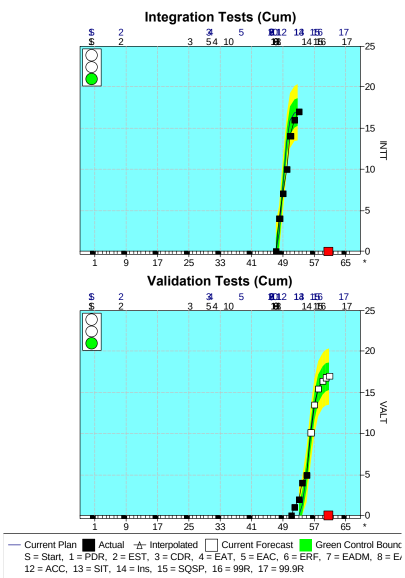


Figure 6: Planned and Actual Integration and Validation Tests

Post Implementation Review- Keeping and Using the History

Keeping and using the basic data described here provides visibility and control throughout the development. It also means a complete history is available when the project completes. This history is invaluable to understand how the project performed and to add to a growing database of (in this case) supplier performance.

During development the risk level in the project is summarized by stoplights. Figure 7 shows the risk status recorded and reported month by month. Notes are kept recording the key factors impacting the development each month.

On completion a formal project review is held between the Telecom software acquisition manager, the Telecom end user and the supplier to identify the success factors and the problems. The notes are used to investigate the history in the development and to learn lessons for future projects.

Conclusions: The (Almost) Perfect Project

All went according to the plan until the final validation tests performed in the last two weeks. At this point there was concrete evidence that high reliability would be achieved and delivery from the supplier would be on time with all the functionality required.

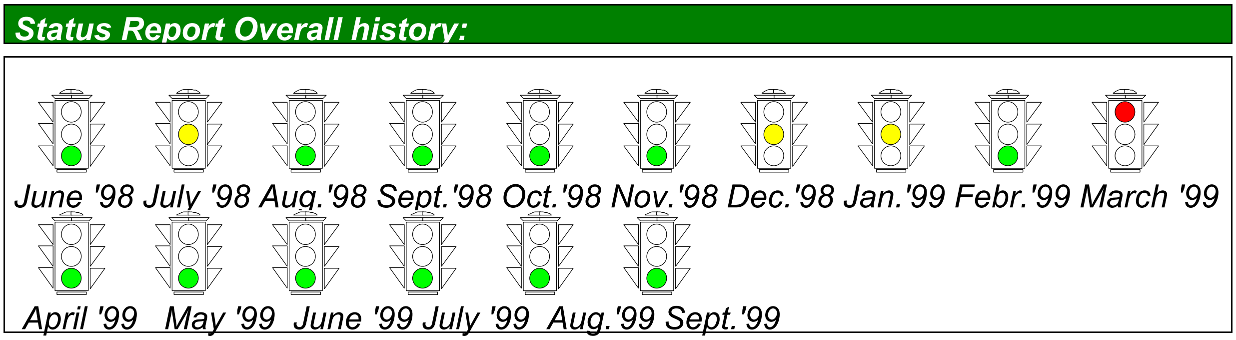


Figure 7: Perfect Project Risk Analysis History

The final validation tests of this complex telecommunications software development include testing the interfaces to network equipment and systems. Unfortunately the Telecom had not assembled one essential set of interface equipment to perform the final validation tests. The result was that completion slipped by 6 weeks due, not to the supplier, but due to the Telecom.

In fairness the Telecom did comment that the project had been amongst the best in their experience. The supplier had kept to the schedule, the budget, delivered all the contracted function and achieved high reliability.

The Telecomm's situation can be contrasted with that found in the FAA. The software purchasing competence in the FAA is assessed based on 6 criteria. [4] The FAA is found to be at risk in every category. Applying the same 6 criteria to assess the Telecom purchasing competence gives the following results.

- **Telecom Corporate Memory:** Suppliers plans are kept and compared with industry reference measures. Over time detailed measures are built on suppliers as their developments complete. These measures are then used to check new proposals from the same supplier and confirm supplier process improvement.

- **Telecom Sizing and Reuse:** Each supplier is formally required to estimate software size including uncertainty and re-use. This size data are used to assess the plan and to quantify the risk. The size data forms part of the contract baseline and are used to track progress in each software module, and control requirement changes.

- **Telecom Extrapolation using actual performance:** The core progress data is used to determine progress against the contract baseline. Variance analysis determines if progress is within agreed limits. If outside the limits then new extrapolations are made of the outstanding time, effort, cost, defects and actual process productivity.

- **Telecom Audit Trails:** The initial baseline plan is recorded together with potential alternatives. All progress data, new forecasts and agreed contractor plans and size revisions are logged

- **Telecom Integrity within dictated limits:** Each supplier proposal is risk assessed against industry reference measures as well as acquisition constraints of time, effort, cost and reliability. Development progress is reviewed continuously to confirm it is within the contract limits.

- **Telecom Data collection and performance feedback:** The development history is captured using the core measures including initial proposal, contract baseline, progress data, forecasts and revised plans. This history is used to continuously update the data repository of supplier performance and highlight those who provide value for money.

Here we see the Telecom motivates suppliers to get and use the SEI core measures to their mutual advantage. This parallels the US Department of Defense motivation in applying maturity assessments on suppliers.

The Telecom is concerned to get commercial benefits from exploiting the SEI core measures. There are real bottom line benefits by using the core measures as show here.

Finally it is a pleasure to describe a real development success. Use of the SEI core measures facilitates success. Too often software case studies [7] [8] are based on disasters, many of which can be avoided by actively using the SEI core measures.

Jim Greene is Managing Director of Quantitative Software Management Europe in Paris, France: telephone 33-140431210; fax 33-148286249. He has over 35 years experience in software engineering, and specialises in introducing management methods based on the quantification of software development for use by development and purchasing organisations. These methods enable CMM Key Process Areas to be satisfied at all maturity levels.

1. Anita D. Carleton, Robert E. Park and Wolfart B. Goethert The SEI Core Measures The Journal of the Quality Assurance Institute July 1994

2. Geerhard W. Kempff “Managing Software Acquisition” *Managing System Development* July 1998 Applied Computer Research Inc. P.O. Box 82266, Phoenix, AZ, USA.

3. Watts S. Humphrey “Three Dimensions of Process Improvement - Part 1: Process Maturity ” *CROSSTALK* The Journal of Defense Software Engineering February 1998.

4. GAO Report to the Secretary of Transportation: Air Traffic Control GAO/AIMD-97-20

5 J. W. E. Greene “Sizing and Controlling Incremental Development” *Managing System Development* November 1996 Applied Computer Research Inc. P.O. Box 82266, Phoenix, AZ, USA.

6. L.H. Putnam “Measures For Excellence: Reliable Software, On Time, Within Budget:” Prentice Hall New York 1992

7. J.W.E. Greene Getting a Runaway Software Development under Control EC2 Software Engineering Conference Toulouse 1990

8. J.W.E. Greene “Avoiding the Premature Delivery of Software” QSM paper see www.qsm.com 1996

For further information on the practices described here, please refer to Lawrence H. Putnam and Ware Myers, Industrial Strength Software: Effective Management Using Measurement, IEEE Computer Society Press, Los Alamitos, CA, 1997, 309 pp.