MEASURES FOR EXCELLENCE

Getting a

"RUNAWAY"

Software Development

Under Control

A Case Study

.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

Development and procurement organisations are often faced with getting a runaway software development under control.

Effective control depends on being confident in the plan being followed and the progress against the plan. For a procurement organisation the need is to evaluate progress independently of the developer. This gives confidence that the expected delivery date and budget can be met. Often very detailed planning is made for a project but this can obscure whether or not the overall plan is feasible.

Detailed planning must govern individual task progress but there is an equal requirement to track progress of the entire team. Analysis is needed each month to determine if the overall progress is consistent with the overall plan. This high level tracking needs to be based on readily available data that does not burden the development team with excessive reporting requirements.

In this paper, we describe the data, measures and techniques to achieve informed management control of an in-progress software development. The technique described operates at the team level. An actual runaway project is used to highlight the practicality of the approach.

The case study project is a telecommunications software development where the contractor originally proposed a fixed price. The project in question can be described as a "worst case" situation. By this we mean an informed evaluation was not made of the original plan. Subsequently a new plan with slipped completion dates and increased costs emerged each time a completion date approached. This typifies a "runaway" project that is out of control.

The need is to evaluate the latest plan and verify if it is realistic and achievable. Straightforward and readily available data is used to assess the plan and the progress. We show the progress is inconsistent with the latest plan. Hence it is necessary to forecast the expected completion date, staffing, cost and software reliability.

2. Objectives

Using the project as a case study, the objectives in applying the measures and techniques described are as follows: -

- demonstrate that essential high level planning data is always available.
- show how the size and development status of the software is determined
- highlight if the plans are consistent with expected industry measures by using the above data.
- quantify progress achieved to determine the development progress each month and overall.
- analyse and determine the variance of this progress data against the latest plan.
- re-plan the overall project to reflect the current position.
- continue to monitor progress against the re-planned project.
- keep management records of what is happening in the development each month.

In summary, the objectives are to demonstrate how a software development is effectively controlled while in progress using readily available data

3. Case Study: Background and Planning History

The project in question is a medium size Telecommunication and Message Switch development. An initial competitive procurement resulted in the contractor gaining the software development contract based on a fixed price.

No evaluation was made of the contractors original development proposal. There were no measures of the process productivity implied by the development plan. Equally no measures were made of the contractors previously completed projects to justify the development proposal. There was a similar lack of quantification of the actual content and size of the development proposed in terms of the individual sub-systems to be delivered.

Contract award for software development began on completion of specification and design.

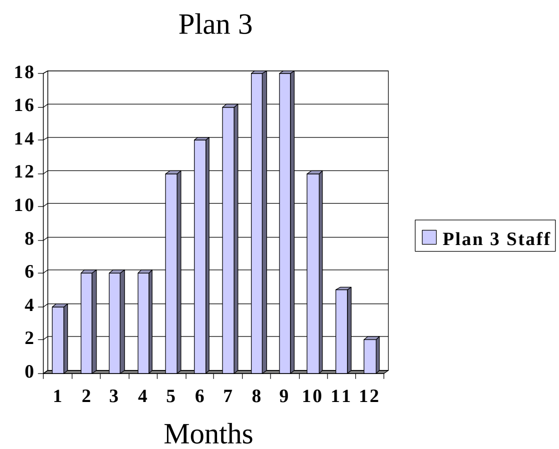
We began by requesting the overall staffing plan for each of the three development plans. A highly detailed Work Breakdown Structure supported each plan. Our interest was the high level summary of the planned staff numbers each month.

3.1 Contract Plan 1

This first plan estimated a development time of seven months involving 59 man-months of effort.

3.2 Contract Plan 2

In the second plan the development time extended to 9 months. Development effort increased to 145 man



3.3 Contract Plan 3

Now the development was planned over 12 months with effort 119 man-months.

By now the procurement organisation had little confidence in the estimating and planning capability of the contractor. Moreover although development had been underway 8 months in all, there was no information on what the contractor had produced in that time or the situation in the project.

We were requested to evaluate the latest plan, determine the current development status, progress and if further slippage and cost overrun could be expected. Of equal concern was the software reliability being achieved and forecast.

To evaluate the series of plans we use the QSM tool SLIM-Metrics that incorporates industry measures from QSM's database of software projects. These reference measures enable plans to be compared with the expected industry values for a given type of application software.

We then use two additional QSM tools to set up the contractor most recent plan and then assess progress against the plan. SLIM, Software Lifecycle Management recreates the most recent plan so that uncertainty in the software product size estimate is reflected. In addition the SLIM output plan enables the software error projections to be established for control purposes.

SLIM-Control captures progress data and uses statistical techniques to determine if the reported high level progress data is within the overall uncertainty limits generated by the SLIM plan. This is called variance analysis. Progress outside the expected limits causes re-planning to determine the completion dates and costs consistent with progress to date.

These three QSM tools generate all the output graphs shown in the remainder of the paper.

4. The Software Development Size and Status

As a first step the contractor was asked to identify the sub-systems under development. Each sub-system was evaluated in terms of the expected size range. The contractor's estimates for the 23 sub-systems are set out below.

Sub-System	Size Range			Sub-System	Size Range		
	Least	Most Likely	Most		Least	Most Likely	Most
1	3500	4300	5000	15	2050	2500	3000
2	3000	3500	4000	16	450	540	675
3	1500	2000	2700	17	500	750	1000
4	2000	2200	2700	18	600	8500	1200
5	3000	4000	5000	19	5000	6500	8000
6	1800	2300	3000	20	400	490	615
7	1500	1800	2200	21	300	400	500
8	3500	4500	6000	22	2800	2820	3000
9	7000	10000	13000	23	10100	11390	14000
10	1500	1990	2560				
11	1200	1800	2500				
12	600	800	1100				
13	510	700	900				
14	520	610	755				

MEAN SIZE = 67,597 +/- 1596

This allows the overall size and uncertainty of the software to be calculated, giving a value of 67,597 +/- 1596 effective lines of code (ELOC).

In addition the contractor was requested to give the status of each sub-system. A simple classification allowed each sub-systems to be identified as being specified, in detailed design, code and unit testing, or under configuration management.

This revealed that, after 8 months in development, only two sub-systems were under configuration management, 21 sub-systems were either in detailed design or being coded.

5. Measuring the Plans

Once the sizing information is returned it is possible to measure and evaluate the three plans using basic data. The measures are made using the two QSM techniques described below.

5.1 QSM Trendlines

Over the last 20 years QSM have assembled a comprehensive database of software project data. This provides up to date reference "Trendlines" of least squares best fits of key development parameters together with standard deviations related to all development phases.

For the three plans we use the reference trendlines for the development phase extracted by analysing the telecommunications and message switch developments in the database.

The results are shown from comparing the three plans against the QSM trendlines for development time and effort.

Our analysis of each plan shows successive plans extending the time to approach the values expected for the size of software to be developed. The corresponding effort analysed in Figure 5.1B reflects the increases in each plan, except Plan 3 where effort reduces. Again these values approach expected values.

However in terms of the timescales and the effort all three plans are assuming less time and effort than industry norms. To achieve such plans implies high process productivity, which we comment on next.

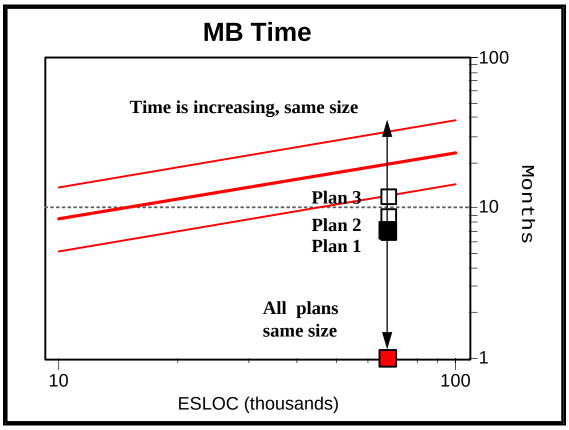


Figure 5.1A Comparison of Planned Main Build (MB) Time

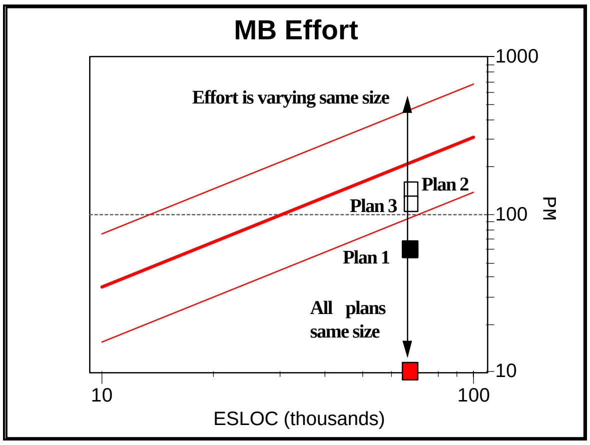


Figure 5.1B Comparison of Planned Main Build (MB) Effort.

5.2 Implied Process Productivity

Each of the three plans gave the time, effort and software size. This enables the calculation of the process productivity required to achieve each plan.

Our measure of the process productivity is in terms of a Productivity Index (PI) expressed using a simple linear scale.

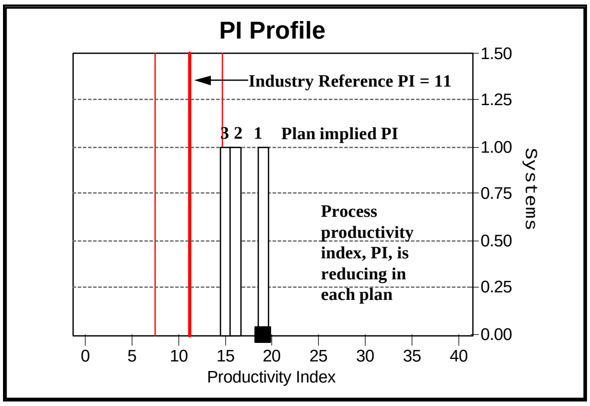


Figure 5.2: Implied PI's for the three plans versus expected industry reference values for telecommunications.

From measures made on the projects in our database we have established that the value to be expected for this type of application software is around 11 to 12. Figure 5.2 sets out the PI values calculated for each plan and shows that each is well above the expected industry values.

6. The Monthly Progress Data

As a first step to determining the current position, the contractor was requested to provide the progress data achieved over the project to date. This basic data was provided and consisted of:

- the development status of each sub-system
- the number of staff working on the project
- achieved major milestones in development
- the amount of code under configuration management
- software errors found in this code.

The next step is to compare this data giving the actual progress to date against the current, third plan.

7. Determining Progress against the Current Plan

By analysing the actual progress data reported against the current plan the variance is determined. This uses statistical quality control techniques to identify when project performance measures fall outside of acceptable bounds. When the progress data is subjected to this variance analysis it is clear that significant deviation is occurring. The two figures 7A and 7B illustrate some of the findings.

Major milestones are shown as the numbers 1 to 7 in Figures 7A and 7B. One major milestone, 2 the critical design review, is reported as completed

To achieve the current plan two further milestones should have been achieved by month 7 namely:

- completion of all coding (Milestone 3 on the charts)
- start of integration test (Milestone 4 on the charts)

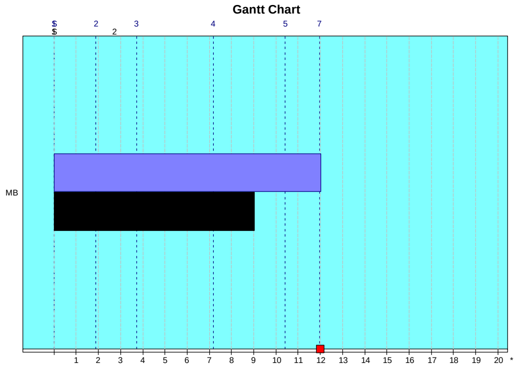


Figure 7A Milestone Variance

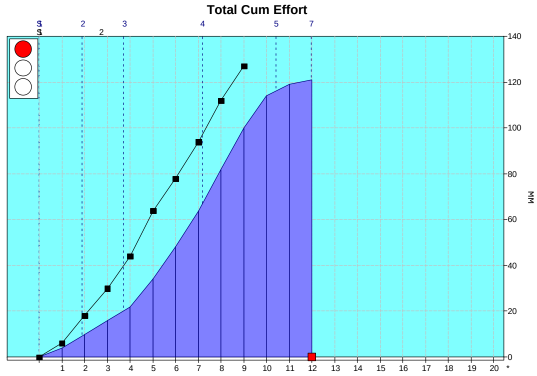


Figure 7B Effort Variance

In fact only some 8,000 lines of code were stated to be complete and under configuration management out of an estimated mean size of 67,597.

Effort variance analysis (Figure 7.B) shows the effort being consumed at the upper uncertainty bound. Thus despite the excess effort progress in the project was well short of the plan.

8. Forecasting Completion Time, Effort and Defects

The re-planning estimate for completing the project consists of three of steps:-

- calculate the productivity index achieved to date using the returned monthly project data.
- determine a weighted overall average of the PI reflecting the degree of confidence for each progress data.
- using the weighted average PI to forecast the schedule, effort and defects for the remainder of the project.

8.1 The PI achieved to date

The weighted PI is calculated using the monthly progress data available so far. An overall weighted value of 11.9 is established. While well below the PI of 15 assumed by the current plan, this is closer to the expected industry norm 11.

8.2 Re-planning Forecasts

The overall weighted PI is used to forecast the monthly expected values to be achieved by the development team. These estimates include outstanding milestone dates, staff levels, lines of code completed and expected numbers of software errors.

Outputs from the forecasts are shown in the following figures that take into account actual progress achieved. A comparison is made against the current plan. Milestones are shown as the numbers 1 to 7 in Figures 8.2 through 8.5.

The new forecast is based on the process productivity achieved to date. Note that the new milestones are forecast and are used to continue tracking progress.

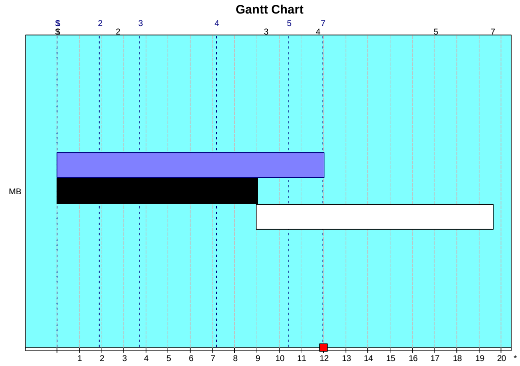


Figure 8.2 Project Gantt Chart: Original Plan versus Forecast

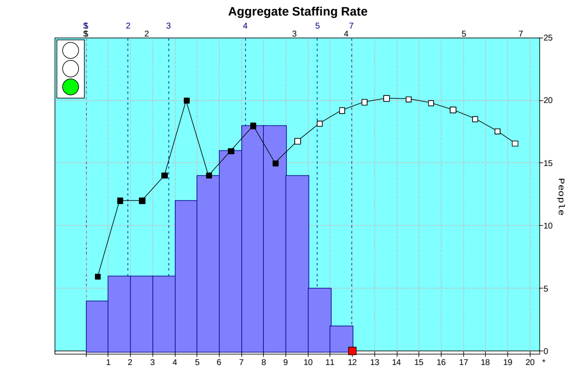


Figure 8.3 Staffing Estimates to Complete

The extra staff effort each month beyond the planned completion in 12 months is substantial. Approximately 18 - 20 staff are required in addition to those planned. At a labour rate of $120,000 per year this means a cost to the contractor of around $200,000 per month.

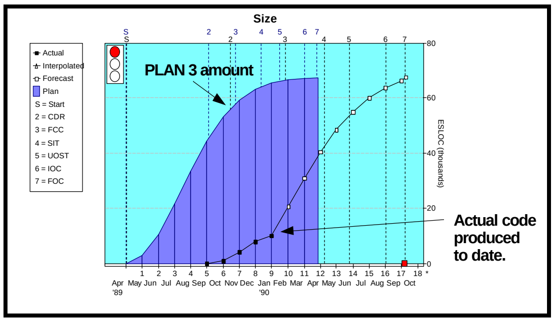


Figure 8.4 Code Generation To-date and Forecast

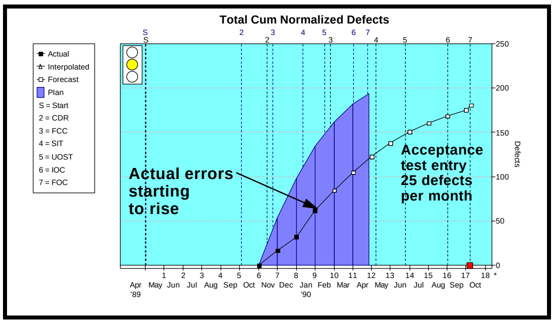


Figure 8.5 Software Errors Forecast

9 Observations and Conclusions

9.1 Observations

Our observations on the situation faced by the contractor and the procurement organisation can be summarised as follows:

- the procurement organisation did not request quantification of the software to be delivered.
- therefore no baseline was established for the end product size.
- equally no measures were made to determine the contractor's productivity either in the plans being put forward or in similar completed projects.
- the contractor produced a detailed WBS that did not reflect the end product size and the development process productivity.
- no evaluation was made by the contractor of the planning constraints (time, effort and cost)
- when requested the contractor is able to quantifying the end product content and size range quickly
- using the QSM trendlines, it is clear that each plan is seriously underestimating the time and effort compared with industry averages
- providing the monthly progress data to date is straightforward and cost the contractor no additional effort to continue to provide each month
- the variance analysis reveals progress to be well behind the most recent plan
- re-planning using the progress data shows substantial additional time slippage, effort and cost overrun will occur
- completion is now forecast for 18 months, a slippage of 6 months on the current plan, 11 months on the first plan
- effort to complete is estimated at an additional 200 man months over the current plan
- there is a need to safeguard against the commercial pressures on the contractor to reduce staff, cut function and deliver before a high reliability is achieved
- the new high level forecast gives a baseline to protect the purchasing organisation against such actions
- re-planning provides new forecasts, including uncertainty bounds, to monitor progress each month.
- forecast of development cost indicates an additional $2,000,000 over the original fixed price
- tracking of software errors is particularly important to achieve high availability at acceptance
- a complete history of the project is captured as each months progress data is recorded.

9.2 Conclusions

Our conclusions are applicable to a growing number of software projects. We find that few software companies have meaningful measures of their process productivity. Without such measures it is not surprising that unrealistic plans are generated.

The content and size of software and its uncertainty must be taken into account separately. The quantification of the size range and hence uncertainty is straightforward and simple to do. The size when analysed with the overall staffing plan, quickly reveals improbable plans compared to expected industry norms. This prevents the procurement or development organisation from accepting or promising the impossible.

In the case study the three high level plans are completely at odds with expected industry values. A high-level viable plan is needed that is consistent with the software size, uncertainty, process productivity and management constraints..

The detailed task plans, in the form of a Work Breakdown Structure need to be produced after establishing the overall viable plan. Detailed task activity plans, while essential to control the project on a day to day basis, are no substitute for using overall measures.

Given a "runaway" project it is practical to get this situation under control rapidly using readily available data. From then on progress is tracked on a monthly or weekly basis against realistic plans that ensure progress is clearly visible.

We recommend all proposed software development proposals be evaluated before they begin. In this way the on-going crisis's that characterises many software projects are prevented.

By performing an informed evaluation of proposals, procurement organisations avoid unrealistic contracts which, while first appearing cheap, in fact cost far more than necessary. The cost to the purchasing organisation due to the delay in installing the system often outweighs the penalty cost in a fixed price contract.

The other benefit is to establish and agree a firm baseline for the software content. Proposed requirements changes are evaluated to determine new completion dates and costs whenever these occur.

Fred Brooks as the Chairman in the US Department of Defense study undertaken to improve software procurement (Ref. 1) concludes that

"Today's major problems are not technical problems but management problems". The task force goes on to call for "*Major re-examination and change of attitudes, policies and practices concerning software acquisition"*.

Watts Humphreys who led the development of the Capability Maturity Model observed (Ref. 2) states that:

"A careful examination of failed projects shows that they often fail for non-technical reasons. The most common problems concern poor scheduling and planning or uncontrolled requirements. Poorly run projects also often lose control of changes or fail to use even rudimentary quality practices."

Our case study highlights how these major problems are overcome by providing managers with high level plans that are checked to ensure they are viable. Each month progress is tracked to confirm delivery of the final software will be on time, within budget and will meet the required reliability levels for acceptance.

Ref. 1: Report of the Task Force on Military Software: US Defense Science Board Sept. 1987.
Ref. 2: Watts S. Humphrey "Three Dimensions of Process Improvement: Part 1 Process Maturity" *CROSSTALK* The Journal of Defense Software Engineering February 1998