Using Line Of Balance To Track The Progress Of Fixing Trouble Reports

© Eduardo Miranda, September 21, 2005

Keywords: testing metrics, project management, tracking and control, Line of Balance, software release management, defect tracking.

Abstract

You are the project manager of a large project, testing is uncovering faults, trouble reports (TRs) are starting to pileup, the release date is coming soon. Are they going to be fixed on time? What could you do to help? Are there any bottlenecks? Where should you assign more resources?

Sounds familiar? Have you been there? This article will explain how could you answer these questions by using an old method called Line of Balance in a new way.

Introduction

Line of balance (LOB) was devised by the members of a group headed by George E. Fouch during the 40’s to monitor production at the Goodyear Tire & Rubber Company. It was also successfully applied to the production planning and scheduling of the huge Navy mobilization program of World War II and during the Korean hostilities. Today LOB application has been further expanded, making it suitable for a whole spectrum of activities ranging from research and development through job shop and process flow operations [1].

In the context of managing a software project, the LOB technique offers two main advantages over the traditional “Open TRs” chart:

- Allows project managers to see, in the middle of a project, whether they can meet the schedule if they continue working as they have been.
- Exposes process bottlenecks; allowing the project manager to focus on those points responsible for slippages.

The “Open TRs” Chart

To answer some of the questions above, project managers usually resort to the “Open TRs” chart shown in Figure 1 or a variation of it.

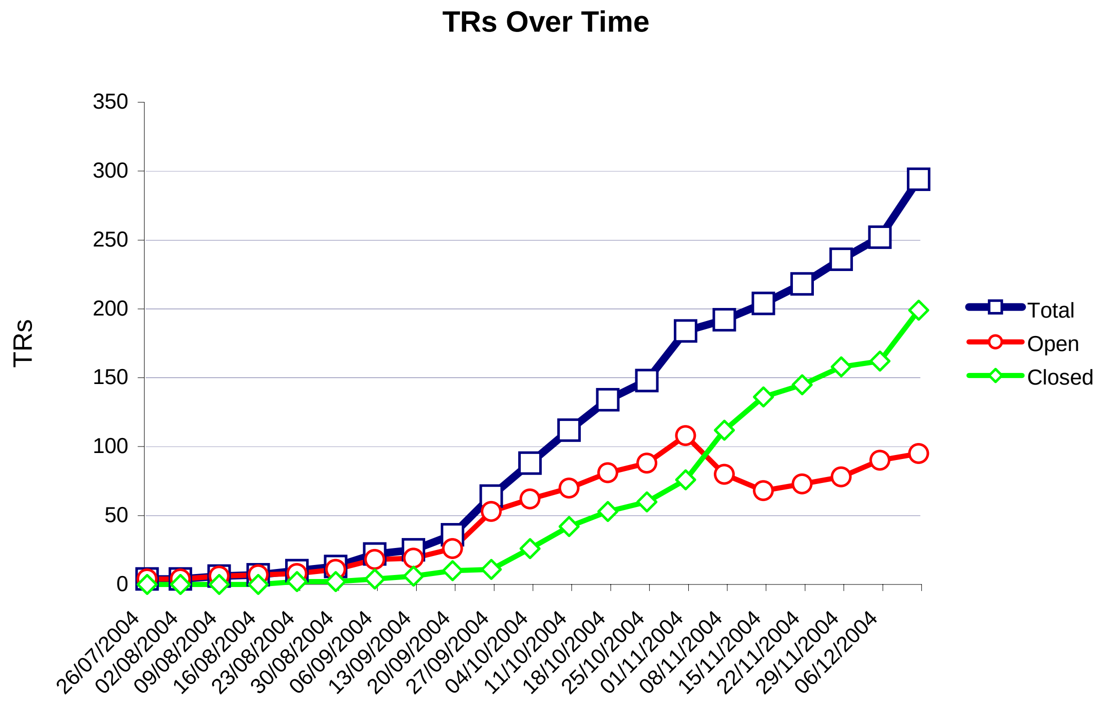


Figure 1 Open TRs Chart

The Open TRs chart shows the cumulative number of trouble reports written over time and its breakdown into open and closed TRs. As the project progresses, the “closed” line should converge towards the total line and the “open” line towards zero. A closed line that is not converging fast enough towards the total or an open line that does not approach zero, signals the project manager the need to shift resources to fixing problems.

Variations of the chart include showing a more detailed breakdown of the TRs status, and ratios between total and open TRs [2, 3].

Despite all its usefulness, the Open TRs chart lacks predictive ability and fails to take advantage of past and present performance data and TRs closure targets; i.e. how many TRs should be in a given state by a given time to meet project deadlines. In other words, although the chart will give the project manager a gut feeling about the situation, it would not answer the questions of where are we in relation to where we were suppose to be? Or how much better should we be doing to get where we wanted to get, by the time we wanted?

The TR Life Cycle

Typically a defect will go through a number of stages or states since it is reported until it is closed. See Figure 2. Each of these states corresponds to a milestone in the process of answering a TR into which the organization or project manager wants to have visibility to evaluate progress, i.e. how many TRs have been reported, how many of the reported TRs have been analyzed, how many of the analyzed were rejected and so on. Elemental states could be grouped onto super sets for reporting purposes, i.e. while the project manager might be interested in how many have been analyzed, assigned, implemented or integrated, the steering group overseeing the project might only been interested in how many TRs were reported, how many were closed and how many still pending.

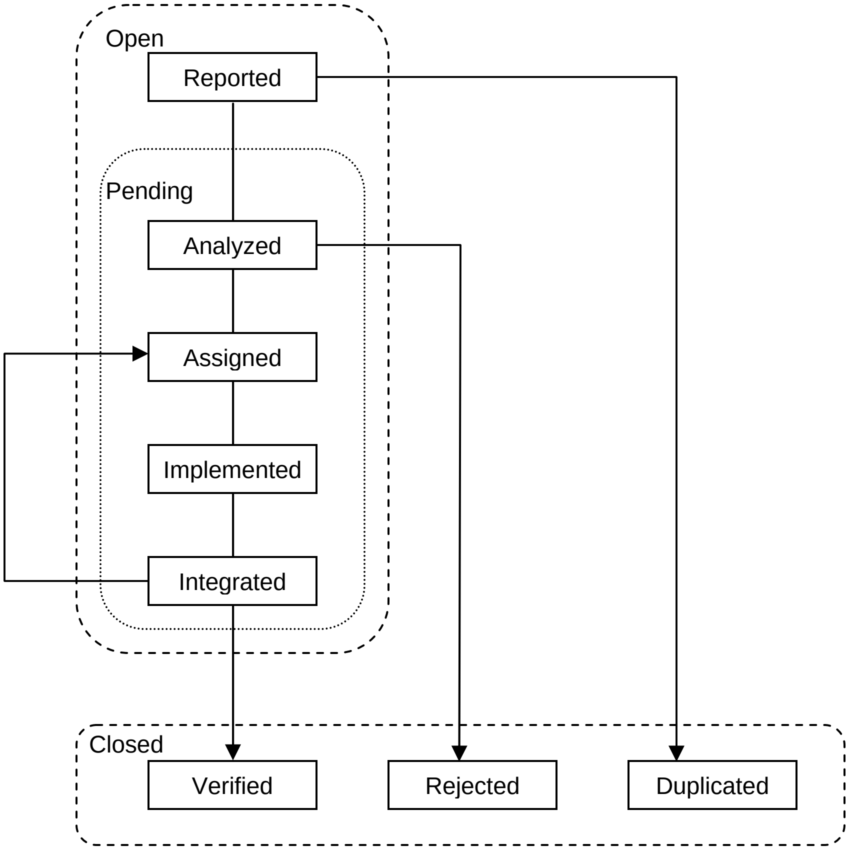


Figure 2 Typical TR life cycle

Most defect tracking systems will implement this or some variation of this model, time stamping each TR as they transition between states. This last feature would allow the organization to produce the lead-time information required by the LOB method.

In addition to the state and timing information, the TR will include other data such as the severity of the problem. This information could be used to filter the TR data and apply the the LOB method to a subset of all the TRs reported and in the prioritization of which TRs to fix first.

The Line of Balance Method Applied to TRs

The LOB method consists of the following elements [4]:

- A number of control points, and their lead times to closing as illustrated in Figure 3, at which progress is to be monitored.

- An “objective” chart or target plan displaying the cumulative closing schedule as planned by the project manager to meet a set deadline. See Figure 4.

- The TR Status chart, which shows the actual number of TRs that have passed through a given control point versus the number that should have been passed (the LOB) according to the plan. See Figure 5.

The information contained in the objective chart, together with the lead-time information is used to calculate how many TRs should be in a given state at a given time.

Control Points

In LOB terminology, a control point is a milestone or event that the project manager wants to monitor. In the context of tracking TRs, the control points and states in the TR life cycle would most likely coincide but this is not necessary[1]. For example the project manager might not find it useful to track TRs in the “rejected” state and so this state would not be considered a control point. The lead-time for a control point is calculated using the following formula:

$$\begin{aligned} &LeadTime_n = 0 \\ &LeadTime_{q \in n-1, n-2, \ldots, 1} = \overline{TimeInState_q} + LeadTime_{q+1} \end{aligned}$$

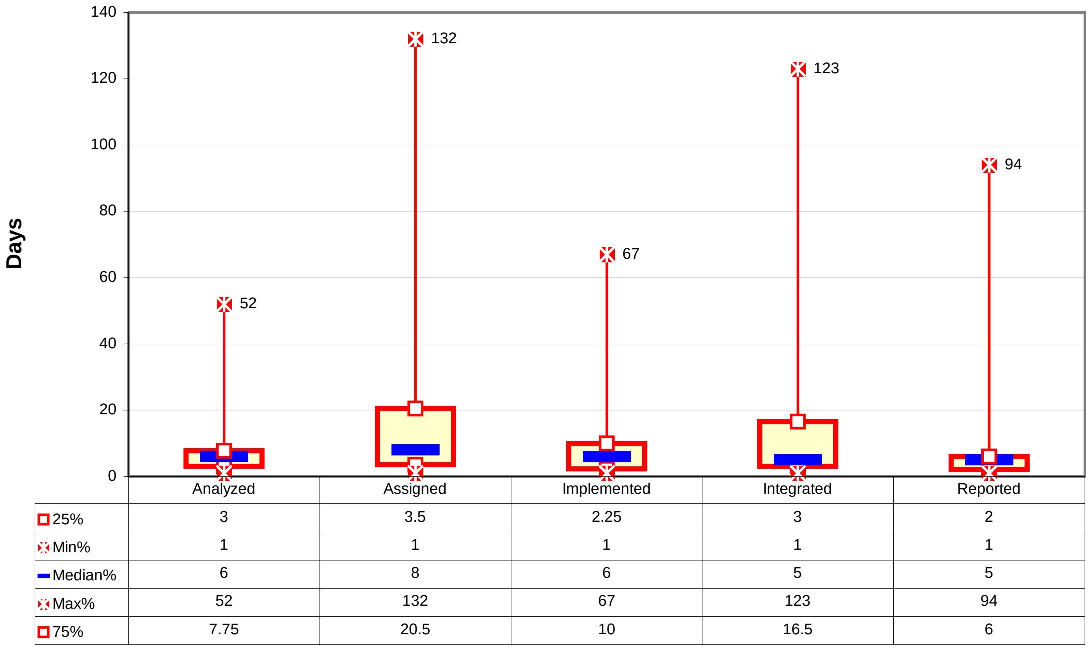
The time spent by a given TR in a given state is calculated according to the formula below:

$$TimeInState_q = \begin{cases} \text{if } TransitionDate_{q+1} \textit{ exists} \text{ then} & TransitionDate_{q+1} - TransitionDate_q \\ \\ \text{otherwise} & CurrentDate - TransitionDate_q \end{cases}$$

[1] The control points are likely to be a subset of the TR states. It is not recommended to create additional control points to avoid confusion.

The $\overline{TimeInState_q}$ could be the average or the median of the times spent by the individual TRs in $State_q$. See Figure 3.

Time In State For Project X As of December 15th, 2004



	Analyzed	Assigned	Implemented	Integrated	Reported
□ 25%	3	3.5	2.25	3	2
☒ Min%	1	1	1	1	1
▬ Median%	6	8	6	5	5
☒ Max%	52	132	67	123	94
□ 75%	7.75	20.5	10	16.5	6

Figure 3 Box-plot chart showing the distribution of times spent in each state by the TR population

Since the time spent in the "assigned" state, see Figure 2, correspond to the time it takes a TR to go from the "assigned" to the "implemented" state, the chart above tell us that the median[2] time it takes to implement a change in the code once the TR has been analyzed is 8 days and that 75% of them take under 20.5 days.

The number at the top of each box plot indicates how many TRs were used on the computation of the statistic. These numbers provide an indication of the reliability of the statistic. A median based in a few observations (TRs) should not be considered very reliable.

Table 1 below, shows the lead-time calculations for each state based on the data extracted from the Time In State chart above. See also Figure 4.

[2] The median is selected instead of the arithmetic mean (average) to prevent rare but very complex TRs from skewing the value of the statistic to the right.

Table 1 Lead-time calculations

Control Point	Time in State	Lead Time
Reported	5	5 + 25 = 30
Analyzed	6	6 + 19 = 25
Assigned	8	8 + 11 = 19
Implemented	6	6 + 5 = 11
Integrated	5	5 + 0 = 5
Verified	0	0

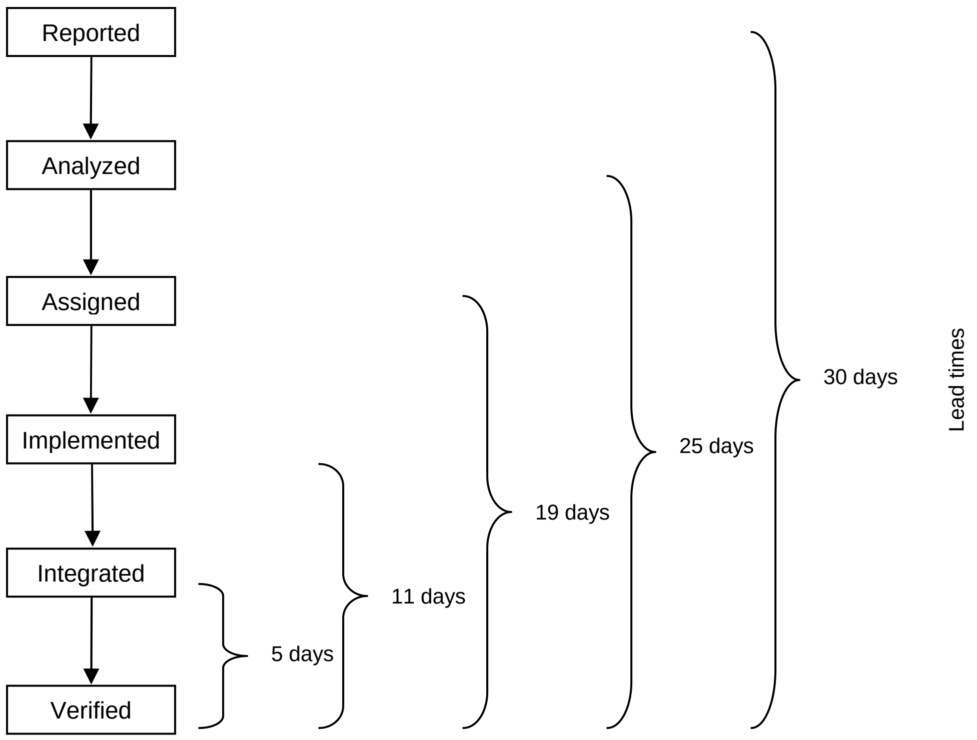


Figure 4 The process of solving a TR and its corresponding lead times

The Objective Chart

The Objective chart, see Figure 5, shows cumulative, "to be verified" TRs on the vertical scale and dates of achievement along the horizontal scale. The chart might also include a display of the achievements so far.

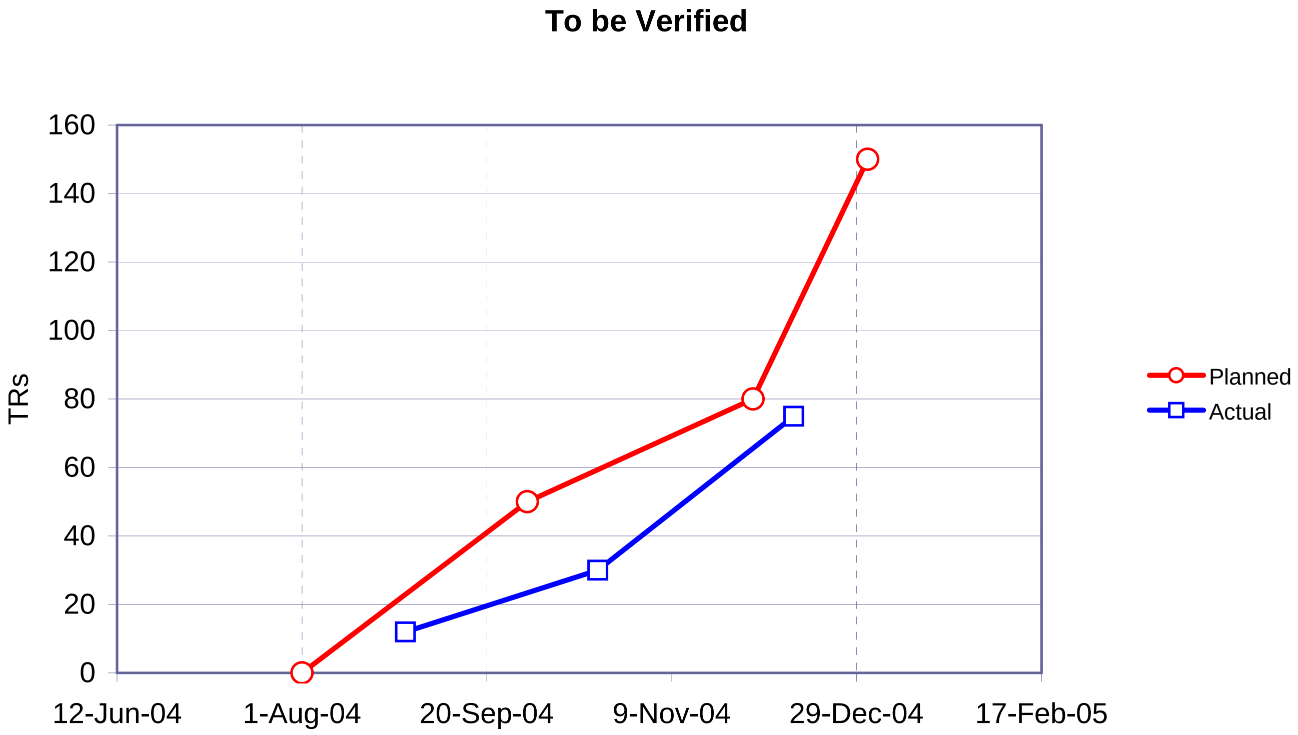


Figure 5 The plan proposed by the PM to clear the TR backlog

The objective chart above shows that the project manager has committed to close 50 TRs by the end of September, 80 by the end of November and 150 by the beginning of the following year. The chart also shows, that as of mid December progress is slightly behind with the project delivering around 75 fixed TRs instead of the 80 promised.

TR Status Chart

The TR Status Chart, see Figure 6, provides quantitative information with regards to progress and whether or not there is a bottleneck on the process.

The chart portrays the actual number of TRs that have passed through each control point against the number that should have been passed according to the plan. These last quantities are called the line of balance. The difference between the LOB and the top of the bar for each control point is the number of TRs behind or ahead of schedule.

Notice that the shape of the LOB will change daily, even if there are no new TRs reported, since its calculation depends on the planned curve of the objective chart and the status date.

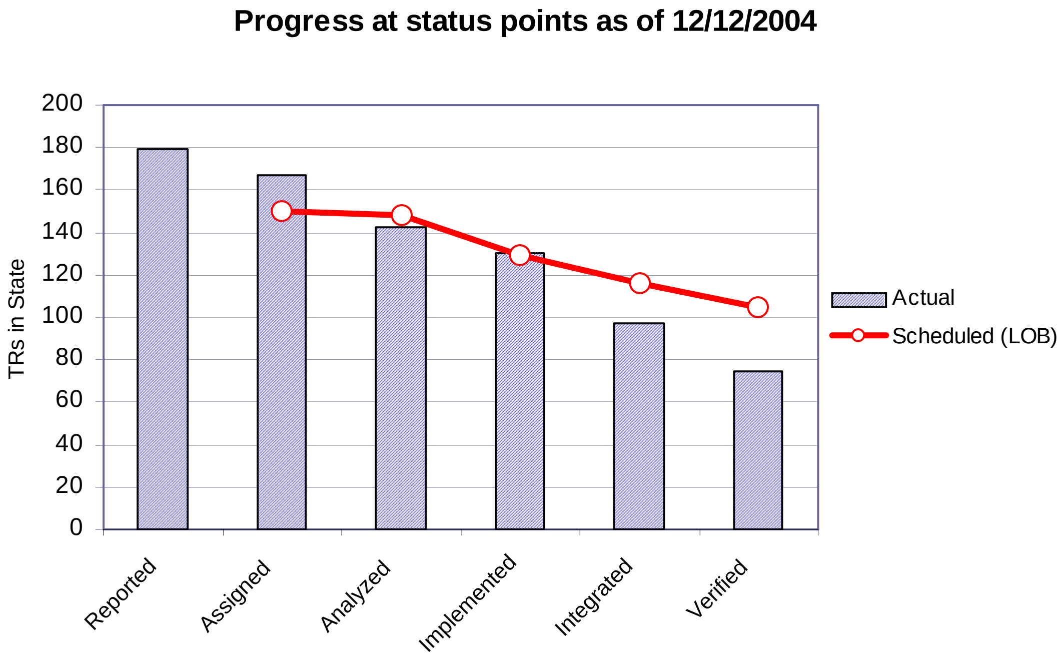


Figure 6 TRs status chart

The TR Status chart above shows that there are almost 180 TRs reported so far, 30 more than what were planned to fix according to the objective chart. This signals the need to update the plan. It also tells us that TR implementation is on track; as the actual column and the LOB line for that control point coincide, but that we are falling behind in their integration and verification. This suggests, that adding more people to implementation activities will not help recoup the delay but that additional resources could be used in integration and verification activities.

The LOB for each control point is calculated as follows:

$$LOB_{q\in 1,2,..,n} = \begin{cases} a_1 + b_1 t & t_1 \le t < t_2 \\ a_2 + b_2 t & t_2 \le t < t_3 \\ . & \\ . & \\ . & \\ a_m + b_m t & t_m \le t < t_{m+1} \end{cases}$$

Where

$b_i = \dfrac{y_{i+1} - y_i}{t_{i+1} - t_i}$ y_i and $y_{i+1}$are the number of TRs to be fixed by time t_i and t_{i+1} respectively, as planned by the project manager and captured in the objective chart.

$a_i = y_i - b_i t_i$

$t = TimeNow + LeadTime_q$

The idea behind the procedure is simple. If it takes on average 10 days for a TR to go from a given state to the completion state, today's status for that state should be equal to the number of TRs that would have to be completed according to the plan 10 days from now. See Figure 7 for a graphical example.

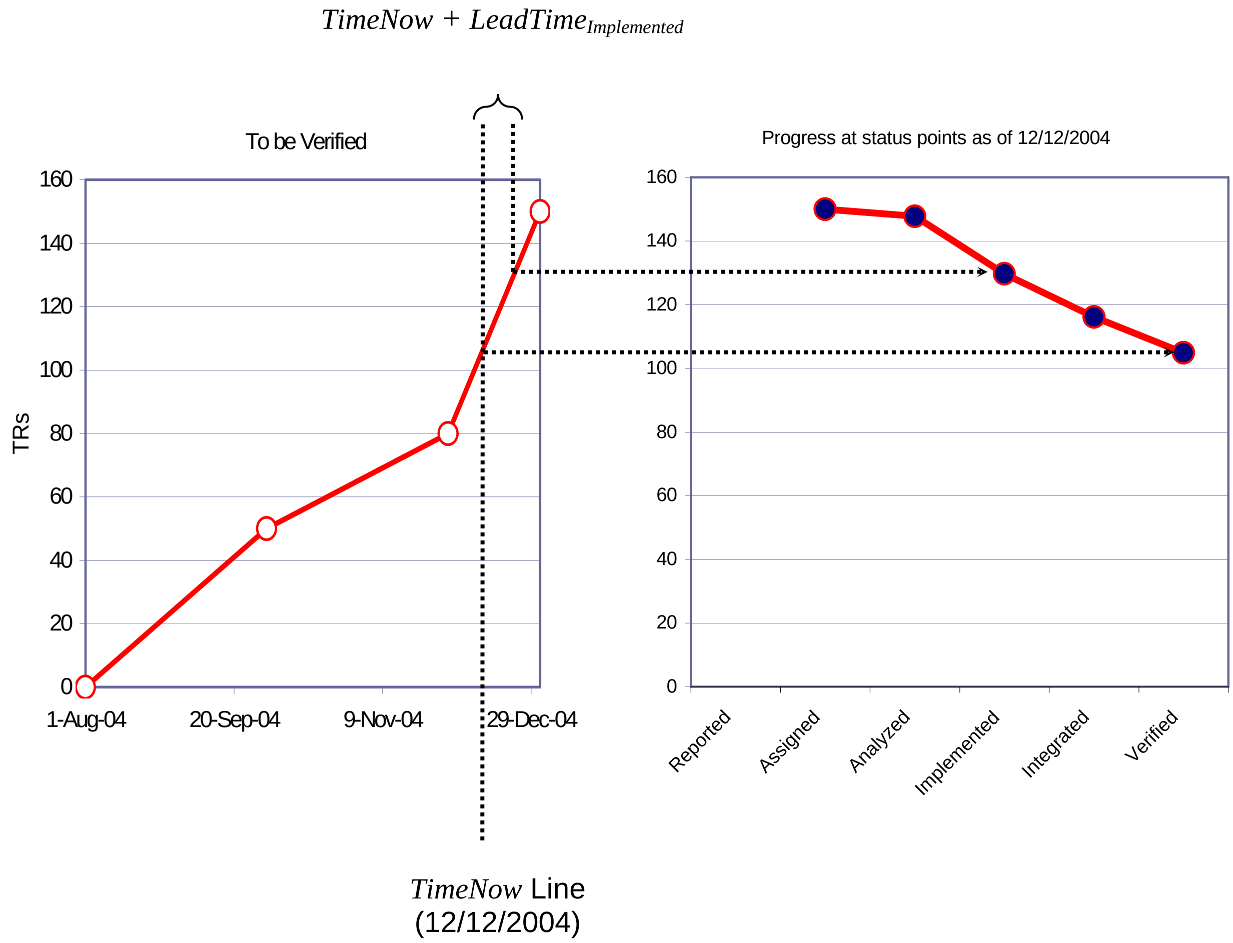


Figure 7 The interception between the *TimeNow* line and the objective chart (the $a_1 + b_1 t, a_2 + b_2 t, \ldots, a_m + b_m t$ function above), yields the value for the Verified control point, that is the number of TRs that should be on that state as of December 12th, 2004. The interception between the same segment and the line at $TimeNow + LeadTime_{Implemented}$ yields the LOB value for the Implemented control point.

Summary

By providing a credible early warning about bottlenecks in the process of fixing TRs, the LOB method helps project managers take corrective actions such as allocating more resources or prioritizing the work when there still time to do it.

In terms of the data required to implement the LOB technique, most of it should be readily available from your defect tracking system or could be derived from it with a few calculations implemented in Excel or any other spreadsheet.

Acknowledgements

Thanks to Jeremy O’Sullivan and Gaetano Lombardi from Ericsson, Alain Abran from École de Technologie Supérieure - Université du Québec and Raul Martinez from RMyA for their comments on earlier versions of this article and to John Corcoran from Ericsson for the TR statistics.

References

1. Line of Balance, Noel N. Harroff, www.nnh.com, valid as Jul 12th, 2005

2. Software Quality Measurement: A Framework for Counting Problems and Defects, William A. Florac, Technical Report CMU/SEI-92-TR-22, Software Engineering Institute, 1992

3. Running The Successful Hi-Tech Project Office, Eduardo Miranda, Artech House, 2003

4. Scheduling for Program Managers, Defence Systems Management College, 2001, http://www.dau.mil/pubs/gdbks/scheduling_guide.asp valid as of Sept. 2005.

Author Biography & Contact Information

Mr. Miranda is an independent consultant. His work spans the development and maintenance of real-time and information management systems. Currently Mr. Miranda is working in the development of new estimation and planning approaches for R&D projects. Mr. Miranda is also affiliated with the Université du Québec à Montréal as an Industrial Researcher.

Mr. Miranda holds a Master of Engineering degree from the University of Ottawa and a Master degree in Project Management from the University of Linkoping. Mr. Miranda has published over ten papers in software development methodologies, estimation and project management and is the author of the book "Running the Successful Hi-Tech Project Office" published by Artech House in March 2003.

Mr. Miranda can be reached at

emt.miranda@computer.org

8500 Decarie Blvd.

Montreal, Quebec

Canada H4P 2N2