Using Line of Balance to Track the Progress of Fixing Trouble Reports

Eduardo Miranda
Independent Consultant

You are the project manager of a large project and testing is uncovering faults, trouble reports are starting to pile up and 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? Does this scenario sound familiar? Have you been there? This article will explain how you can answer these questions by using an old method called Line of Balance in a new way.

Line of balance (LOB) was devised by the members of a group headed by George E. Fouch during the 1940's to monitor production at the Goodyear Tire & Rubber Company [1]. 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.

In the context of managing a software project, the LOB technique offers two main advantages over the traditional *Open Trouble Reports (TRs) Chart* [2]:

- It 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.
- It exposes process bottlenecks, allowing the project manager to focus on those points responsible for slippage.

The Open TRs Chart

To answer some of the questions raised at the beginning of this article, project managers usually resort to the Open TRs Chart shown in Figure 1 or a variation of it.

The Open TRs Chart shows the cumulative number of TRs written over time, and its breakdown into open and closed TRs. As the project progresses, the closed line should converge toward the total line and the open line towards zero. A *closed* line that is not converging fast enough toward the total or an *open* line that does not approach zero signals to the project manager the need to devote additional resources to fix problems.

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

Despite all its usefulness, the Open TR 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 the 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 are suppose to be, or how much better we should be doing to get where we want to get by the time we want.

The TR Life Cycle

Typically, a TR will go through a number of stages or states since it is reported until it is closed (see Figure 2, page 24). 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 into 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 were still pending.

Most defect tracking systems will implement this model or some variation

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

The LOB 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 (see page 24), 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 (Figure 4).
- The TRs Status Chart (see Figure 5, page 25), 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.

The information contained in the
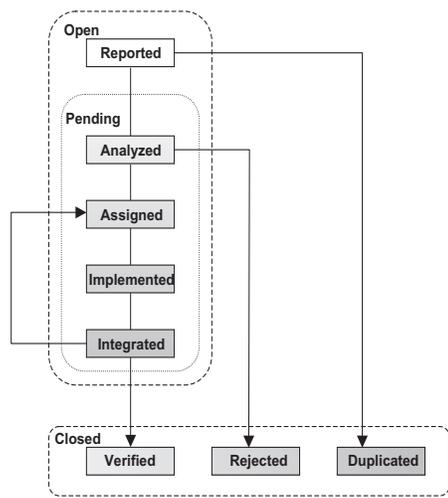
Figure 1: *Open Trouble Reports Chart*
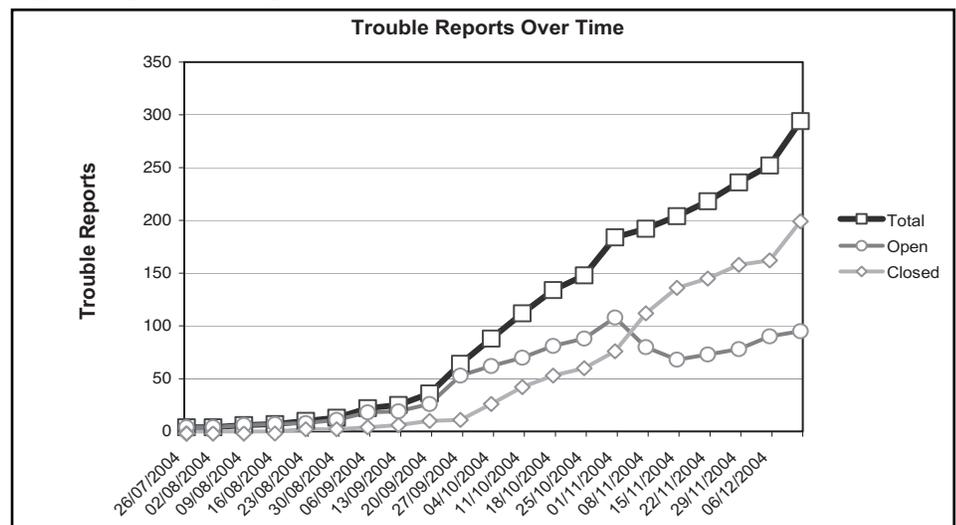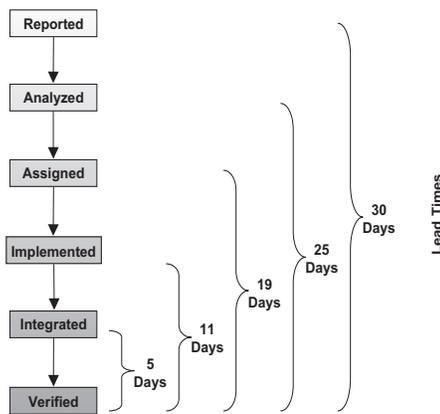
Figure 2: *Typical TR Life Cycle*

Figure 3: *The Process of Solving a TR and Its Corresponding Lead Times*

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

Table 1: *Lead-Time Calculations*

Figure 4: *The Plan Proposed by the PM to Clear the TR Backlog*

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:

$$LeadTime_n = 0$$
$$LeadTime_{q=n-1,n-2,...,n-1} = TimeInState_q + LeadTime_{q+1}$$

Assuming that the median[2] times a TR spends in a given state are those shown in Table 1, the lead-time calculations will yield the results illustrated in Figure 3.

The Objective Chart

The Objective Chart 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.

The Objective Chart in Figure 4 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 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 LOB. 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.

The TR Status Chart 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:

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

$$b_i = \frac{y_{i+1} - y_i}{t_{i+1} - t_i}$$

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 an average of 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 6 for a graphical example.

In Figure 6, the chart on the left shows the planned line from Figure 4, while the chart on the right shows the scheduled

line from Figure 5. We obtained Figure 5's scheduled line by finding the interception between the *TimeNow* line and the curve in the objective chart (the $a_1 + b_2 t, ..., a_m + b_m t$ function above), which yields the value for the Verified Control Point, that is the number of TRs that should be on that state as of Dec. 12, 2004. The interception between the curve 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 is 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. Harroff, Noel N. "Line of Balance." NNH Enterprise, 7 June 2003 <www.nnh.com>.
2. Pussacq Laborde, Juan. "Quality Control = Project Control?" Second Software Engineering Process Group Latino America Conference. Mexico, 2005 <www.esi.es/SEPGLA/index_eng.html>.
3. Florac, William A. "Software Quality Measurement: A Framework for Counting Problems and Defects." CMU/SEI-92-TR-22. Pittsburgh, PA: Software Engineering Institute, 1992.
4. Miranda, Eduardo. Running The Successful Hi-Tech Project Office. Artech House, 2003.
5. Defense Acquisition University. Scheduling for Program Managers. Defence Systems Management. College Press, Oct. 2001 <www.dau.mil/pubs/gdbks/scheduling_guide.asp>.

Notes

1. The control points are likely to be a subset of the TR states. To avoid confusion, do not create additional control points.
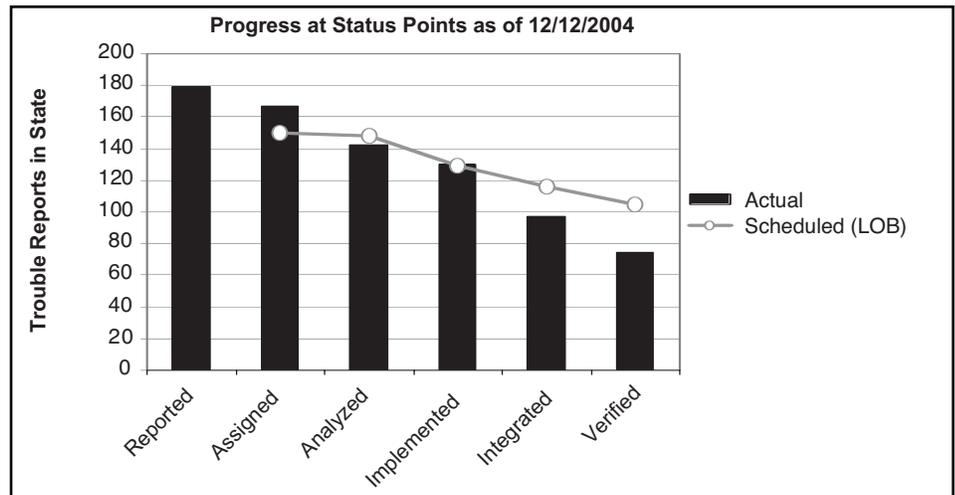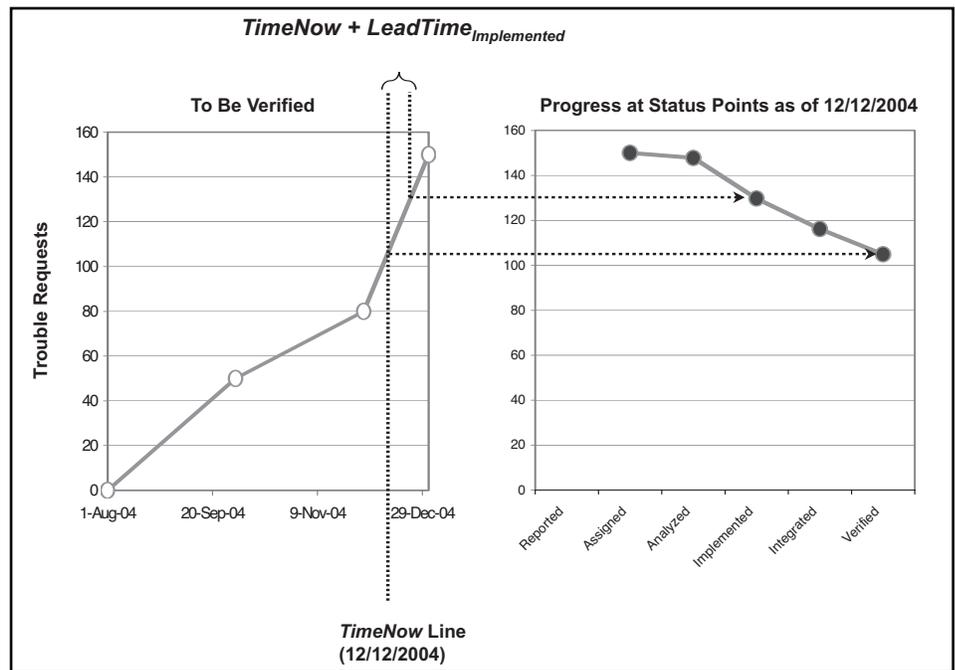

Figure 5: *Trouble Reports Status Chart*

Figure 6: *The LOB for the Control Point*

2. The median is preferred to the arithmetic mean (average) to prevent rare but complex TRs from skewing the value of the statistic to the right.

About the Author

Eduardo Miranda is a system professional with 20 years of experience in the development of software-based products and information management systems. Currently, he works in the development of new estimation and planning approaches for research and development projects. Miranda is affiliated with the Université du Québec à Montréal as an industrial researcher, and is a member of the International Electronics and Engineers. He has published more than 10 papers in software development methodologies, estimation, and project management and is the author of "Running the Successful Hi-Tech Project Office." Miranda has a Master of Engineering degree from the University of Ottawa and a master's degree in project management from the University of Linkoping.

119 Harwood Gate
Beaconsfield, Quebec
Canada H9W 3A5
Phone: (514) 697-0594
E-mail: emt.miranda@computer.org