Setting Control Bounds in SLIM-Control

Kate Armel and Paul Below

When he created control charts in the 1920's, Walter Shewhart was concerned with two types of mistakes:

- Assuming common causes were special causes
- Assuming special causes were common causes

Since it is not possible to make the rate of both of these mistakes go to zero, managers who want to minimize the risk of economic loss from both types of error often use some form of Statistical Process Control.

The control bounds in SLIM-Control perform a related, but not identical function. Control bound thresholds operate as a trigger. When a given control threshold is crossed, SLIM-Control alerts managers whenever the *actual values* of a given metric deviate far enough from their *expected (or planned) values* to merit investigation and possible corrective action.

Of course, not all deviations from planned (or "expected") values are cause for alarm. After all, a development team who consistently outperform their peers may find their actual code counts consistently exceed a conservative project plan. But unless their work is bug-ridden, this "different than expected" performance should be cause for celebration rather than alarm.

Similarly, the meaning of higher than expected defect discovery rates is not always clear. Since defect discovery is a function of two variables (defect density and time spent testing/efficiency of testing), it may not immediately be apparent which variable is driving higher than expected defect counts. The testers may be doing a superlative job, the product may in fact be buggy, or high defect counts may signal that both things are going on. For this reason, a yellow or red traffic light should not automatically be viewed as an ominous sign. The sole function of control bounds is *to bring the metric to the attention of management when a specified variance from the plan occurs*. Once alerted, project managers can investigate the cause of this "different than expected" performance, assess the likely causes and possible impact on the project plan, and decide whether corrective action is required.

So: *how much* variation between the plan and actual values is acceptable? How narrow or wide a span of control is optimal?

Many users think the control bounds ought to represent a sort of "sigma" or industry average range of variation from industry norms. But is that really what we want control bounds to do? After all, the amount of variation in industry defect data gathered from a wide range of firms, all of which use different defect definitions and testing procedures, and who report different combinations of severity categories has little to do with *a particular development manager's subjective risk tolerance on a particular project.* We might expect a narrower span of control (and lower risk tolerance) for mission

critical applications that deliver high reliability around the clock. Likewise, a broader span of control (and higher risk tolerance) for IT applications which only run 40 hours a week may more appropriately balance risk with efficient resource allocation.

Because determining an appropriate span of control is not always an easy task, SLIM-Control ships with a default set of control bound rules based on over 20 years of experience with tracking defect data (and other metrics) across a wide range of environments. These "default" control limits can be adjusted to suit both the organization's subjective risk tolerance *and* the variability inherent in that organization's data.

At what point should you consider corrective action or adjust the project plan?

When statistical control thresholds are exceeded we don't always know the correct cause, let alone the correct response. There are four possible responses to such a warning. Two are appropriate and helpful (green). The other two responses (in red) do nothing to improve performance:

Tracking against Control Bounds	**Actual Situation**	
	Not a problem	**Real Problem**
Action Triggered	**Wasted effort researching and replanning**	**Correct result**
No Action Triggered	**Correct result**	**No action taken, problem worsens**

QSM's default control bounds are intended to provide a reasonable balance between these two types of errors. They can be customized to broaden out or narrow the span of control as the delivery date approaches or to alert you only when metric values stray too far above or below the plan.

For additional reading, refer to the *IEEE Computer Society Executive Briefing: Controlling Software Development* by Larry Putnam Sr. and Ware Myers, 1996. Specifically, Chapter 7 which is on Monitoring Project Progress.