

Sizing Up the Peer Review — Real Metrics in Real Time

by Mr. James Gonzalez, Software Specialist, DCMA Boeing Long Beach

A valuable metric for a DCMA software quality assurance representative is the number of peer review defects found in a contractor's software code or software review documentation. But just how valuable is this metric? And how many defects are too many, too few or just right? Do you have to wait and compare data from different time periods before the metric is useful? I finally found the answer to these questions — the trick is using the peer review data in a different way.

As a DCMA software quality assurance representative working with an acquisition category (ACAT) I program, the C-130 Avionics Modernization Program, I had been trained that peer review metrics — the number of defects, both major and minor, found during peer reviews — are part of every software

engineering program. The program I am working with was no exception, and the contractor had included a battery of such metrics for cost, schedule, issues closed, etc. Prior to a delivery of software code to the customer, the contractor published new peer review defect data. I eagerly reviewed the data, wanting to gain insight into the code's quality. But I realized I had no idea what the number of defects meant — there was no context for the data. Was this too many defects? Too few? When I searched for an industry standard for an acceptable number of defects, I found that there really wasn't one. How could there be when every new acquisition program is unique?

When I queried the contractor personnel as to how they used the metric, they replied that as the software went through the development phases, they would compare the number of defects from one phase to the next.

If the new code had more defects, that would indicate that it was problematic. Though this explanation

I had been trained that peer review metrics — the number of defects, both major and minor, found during peer reviews — are part of every software engineering program.



(Above) Mr. James Gonzalez, DCMA Boeing Long Beach software quality assurance representative, outside a large model of a C-130 cockpit, the design of which his peer reviews positively impact. (DCMA staff photo)

My new approach was to use the peer review metric to judge the stability of the peer review process itself, rather than judge the software code.

made sense on the surface, I was still puzzled. If they found fewer defects in the most recent development phase, did that, in fact, mean the code was better — or could it mean the peer review process was not as effective in that phase? And how similar was the new phase from the previous one? By its very nature it must be different in some way — was the development in one phase more complex? And what about the team makeup — had it changed through attrition or reassignments?

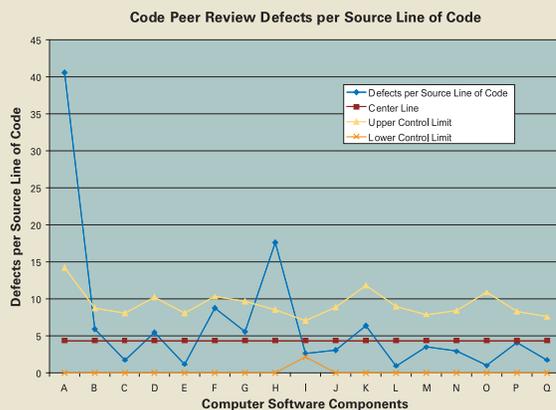
The traditional peer review metrics began to look less and less straightforward, and I began to doubt their accuracy and wonder about their usefulness. But after some research, I discovered that the peer review data is

The use of the statistical chart as a metric for peer reviews revealed costly problems with the contractor's peer review tool and its employees' training that, unchecked, would have resulted in years of erroneous data.

(U chart), would reveal the stability of the process, provided there were 10 to 20 separate peer reviews conducted, each covering a single, large piece of code or document. With that many data points plotted on the chart, the behavior of the peer reviews, i.e., the number of defects found, should all fall within the norm of a statistical bell curve.

However, when we plotted the number of defects found in recent code peer reviews, we discovered that five out of 17 peer reviews showed the number of defects to be outside the allowable limits. The interesting aspect of this metric is that it cannot point to the specific

problem, but it can direct the user where to look. It was now up to the process owners to investigate why those particular peer reviews found too many or too few defects.



important; after all, it is the real output from a long, expensive and resource-intensive process.

My new approach was to use the peer review metric to judge the *stability* of the peer review process itself, rather than judge the software code. The metric, if plotted in a statistical chart

When the process owners conducted their root-cause analysis, they discovered a major problem with the tool that recorded the defects as well as problems with how the reviewers were trained. The tool, which is a spreadsheet that records each error, used a formula that was incorrectly counting enhancements as defects. Compounding this problem, reviewers were incorrectly labeling enhancements and failing to complete other pieces of data. In the case where one peer review had too few flaws, the evaluation showed that the section of code reviewed was largely composed of auto-generated code. The auto-generated code artificially increased the total number of lines of code, thereby reducing the average number of defects. After the process owners corrected these issues, they charted the

(Above) The peer review metric, if plotted in a statistical chart, reveals the stability of the peer review process, providing that 10 to 20 separate peer reviews are conducted, each covering a single, large piece of code or document. With that many data points plotted on the chart, the behavior of the peer reviews, i.e., the number of defects found, should all fall within the norm of a statistical bell curve.

For DCMA, this peer review metric is a tool to evaluate the health and stability of the contractor's peer review process and help contractors manage their resources more effectively to produce a quality product for our customers, on time and within cost.



data again and determined that only two of the peer reviews showed defects above the allowable limits.

Focusing on these two peer reviews, the owners determined that the developer for one section of code did not have the right kind of experience and thus was reassigned to more suitable work. They also realized that the other developer needed more help, and they rotated in additional support resources.

The use of the statistical chart as a metric for peer reviews revealed costly problems with the contractor's peer review tool and its employees' training that, unchecked, would have resulted in years of erroneous data. The metric guided the contractor's management personnel to evaluate their employees' performance and thereby improve the production of their code. This all was possible by using the statistical nature of the metric to compare the number of defects against similar peer reviews covering similar code.

(Top) Mr. Gonzalez's peer review metric impacts the design of the C-130 cockpit, shown here, which is being upgraded through the C-130 Avionics Modernization Program. (DCMA staff photo)

Even though I had suggested the statistical measurement as an outsider, the contractor immediately adopted it, and they now plot the metric data after each session of peer reviews to ensure a stable process and to verify resources are performing as planned. The immediate acceptance of the statistical measurement process may be partly due to the fact that they already had the data available — applying the chart parameters to the traditional peer review data is

not difficult. And despite some initial suspicion, they were ultimately thrilled with what the new metric was showing them.

The U chart's strength lies in its ability to look at peer review metrics in real time without the need for industry standards. It can realistically compare the results because it is comparing similar processes and similarly trained resources that haven't changed over time.

For DCMA, this peer review metric is a tool to evaluate the health and stability of contractors' peer review processes and help contractors manage their resources more effectively to produce a quality product for our customers, on time and within cost.

The author would like to thank Mr. Ronald J. Weis, the senior technical advisor at DCMA Boeing Long Beach, for championing this article, routing it through the various loops for approval and reviewing it for technical accuracy.