I was once contacted by a major telecommunications company to assist in a study of its software quality assurance methods and measures. The company was perplexed over the confusing trends of error data for a sequence of releases of the same product and wanted an outsider to take a look.

The measure of quality used was defects per KLOC. If 23 defects were found per thousand lines of code, the defect ratio is 23/1000, or 0.023. Often, the number 23 would simply be used because the context is clear.

The company has a standard software development process and had an independent SQA group (though very small).

The company used a software inspection process (patterned after Michael Fagin) to identify and count defects. At the end of an inspection test, a person (the recorder) was responsible for entering results into a database for further analysis. Values entered included number of lines inspected, total time for effort inspection, and number of defects found. Another software system (a software configuration management system) kept track of module sizes (LOC) as the product evolved. Hence, at the end of the development effort, counts of defects discovered and LOC counts were available to compute defects per KLOC.

Sounds great—so why were results confusing?

1. No real correspondence between SCM software and defect software.
2. No reporting of defects during testing.
3. No reporting of LOC size changes after inspections
4. No real SQA procedures followed
5. No certainty that inspectors were trained or knowledgeable or trained
6. No control over data entered into inspection database
7. Management specified defect per KLOC expectation
8. Perceived punitive of software size estimates and reality by management
9. No effective software engineering progress group (SEPG).
10. And more…