STANDARD SOFTWARE QUALITY METRICS

James Inglis

***James Inglis** is supervisor of the Software Quality group in the Quality Assurance Center of AT&T Bell Laboratories, Holmdel, New Jersey. The group develops and implements techniques for measuring and improving software quality. He received a B.A. from Amherst College in 1967 and an M.S. and a Ph.D. in statistics from Stanford University in 1968 and 1973. He joined AT&T in 1978.*

Standard measures of software quality have been set up for AT&T Bell Laboratories. These metrics allow a software project to be followed through its development, controlled introduction, and release to customers. The metrics serve both project and corporate management needs. For project management, they allow more effective management of development effort, and they help to ensure a fast and effective solution to problems that arise at any stage. For corporate management, they provide a vehicle for quantifying the overall quality of software development, for setting quality improvement objectives, and for tracking results. In particular, the metrics provide quantitative information on number of faults, normalized so that corporate results can be summarized and projects of differing size can be compared; the responsiveness of support organizations in resolving problems; and the impact of fixes on customers.

In 1983, AT&T Bell Laboratories established a standard set of metrics, or measures, of software quality for reporting on all software development projects within the company. These metrics were selected for broad applicability across the wide range of software development projects within Bell Laboratories and were based on data that were generally available. Since their introduction, the metrics have undergone revision, but their general purpose remains the same.

Standardized Quality Metrics

The Quality Assurance Center assumed responsibility for generating a semiannual report on projects in terms of the standard metrics. After two prototypes, a transitional pilot report appeared in early 1984 and regular reports have appeared since then. The Software Quality Assurance Report (SQAR) provides information on the quality of AT&T generic software during system test, controlled introduction, and the postrelease period.

Quality Metrics In Use

The standard software quality metrics provide useful information to project and corporate management. The metrics permit the evaluation of trends and the quantifiable analysis of quality, starting with system test. The measurements quantify

- The number of faults in generic software, normalized by software size
- The responsiveness of development and customer support organizations in resolving customers' problems
- The impact of software field fixes on customers.

A description of each of these measurements follows.

Cumulative Fault Density

The cumulative fault density provides a cumulative measure of the known faults, normalized by total software size, contained in a release during its useful life cycle. Two metrics graph cumulative fault density: one shows faults found within the company, the other shows faults found by customers. This makes clear not just how many faults there are, but by whom they were discovered.

The Cumulative Fault Density—Faults Found Internally graph (Figure 1) depicts the faults found by the development organization (developers, testers, and customer support personnel), normalized by the total software size in the system test phase. The graph plots this metric cumulatively starting with system test.

The Cumulative Fault Density—Faults Found by Customers graph (Figure 2) depicts the faults found by customers in the normal operation of released software, normalized by the total size of the released software. Any fault that is identified as a result of a customer-initiated inquiry or complaint is counted as "found by a customer." The horizontal axis is in system-months after release, on a logarithmic scale. Using system-months reflects the differing numbers of customers different software products have and their differing exposure to opportunities for customers to find faults. The logarithmic scale keeps the graph from appearing distorted.

Serious Fault Status Distribution

The Serious Fault Status Distribution graph (Figure 3) reports the number of serious faults found and the status of those faults—open (uncorrected) or closed (corrected)—as of the graph date. This provides a "snapshot" of the current status of serious faults. It gives an indication of how fast the project staff moves on closing faults once they are discovered. This graph is plotted from the beginning of controlled introduction.

The Mean Time to Close and Mean Time Still Open for Serious Faults graph (Figure 4) provides a measure of the responsiveness of the development and customer support organizations by showing the average time that serious faults remain open. The Mean Time to Close curve reports the average time to close, using a three-month rolling average, for serious faults since the start of controlled introduction. The Mean Time Still Open value for each month is the mean length of time that the serious faults open at the end of the current month have been open. Thus, these two measures cover both recently closed faults and those that have remained open.

Field Fixes

The Field Fix Distribution (Figure 5) provides a measure of the impact of software field fixes on customers. A field fix may correct one or more faults and may be distributed in a variety of ways. This graph also shows the number of systems in service.

The Field Fix Applications graph (Figure 6) shows the number of applications of field fixes customers must install. This is basically the product of the two measures, field fixes and systems in service, in Figure 5.

Figure 1. Cumulative fault density—faults found internally.

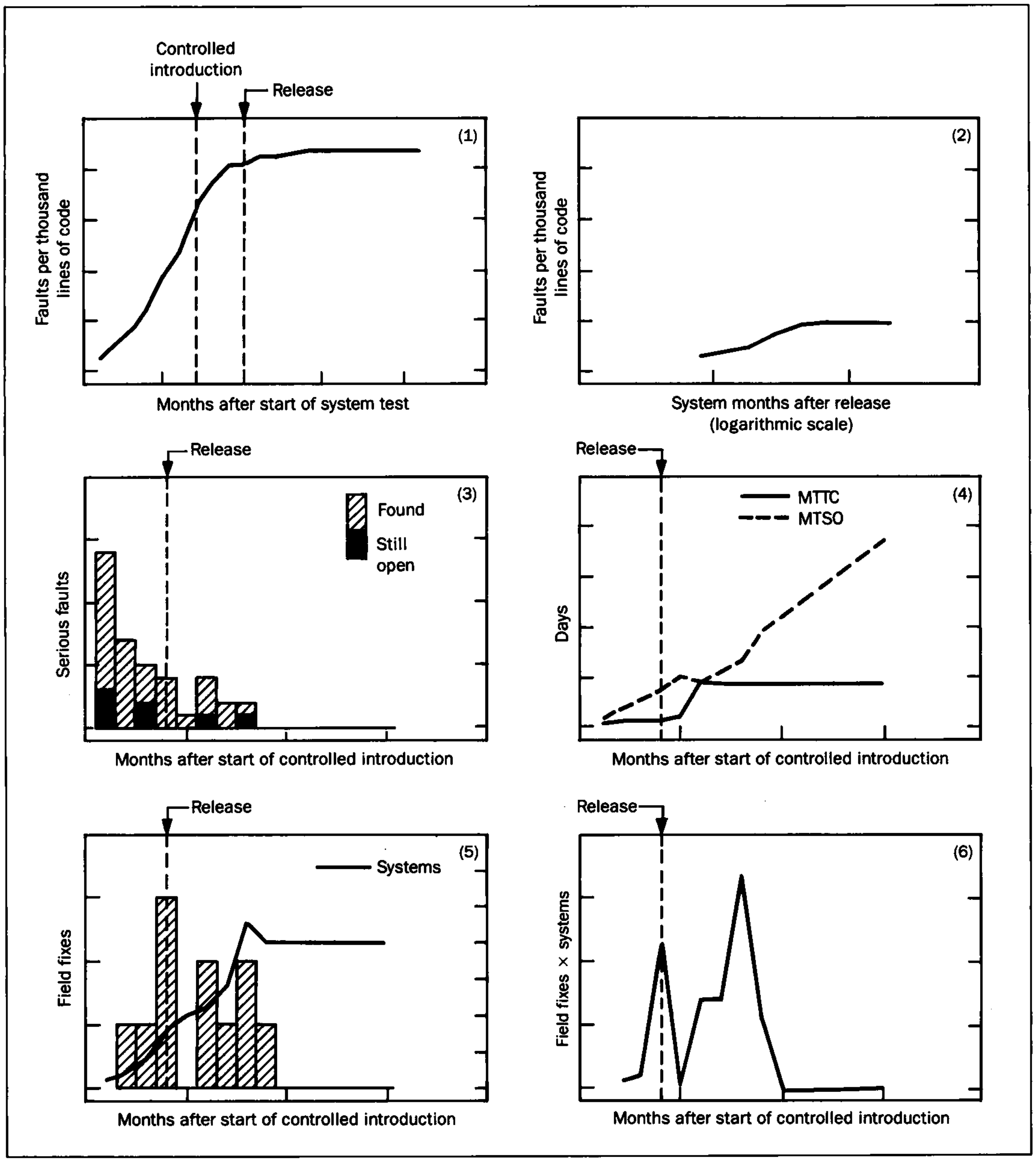
Figure 2. Cumulative fault density—faults found by customers.

Figure 3. Serious fault status distribution.

Figure 4. Mean time to close and mean time still open for serious faults.

Figure 5. Field fix distribution.

Figure 6. Field fix applications.



Metrics Summaries

Standard, broadly applicable metrics provide opportunities for several types of analysis. Two are of particular interest: summarization and comparison. Summarization yields a unified, overall view of software quality for the corporation. It shows where a group of projects are at a given time and where they are headed.

Comparison is equally useful, but must always be done with caution. Because software releases differ in many ways, simple comparisons based solely on the reported software quality metrics may not be appropriate. However, comparison of the metrics for a release with a previous version can indicate in a quantifiable, objective way what improvement (if any) has taken place. Comparison of the metrics for projects with different development methodologies, different development environments, or different management procedures can provide information on relative advantages and disadvantages.

The summary's purpose is to present an overview of selected values from the results for an individual software release. There are many ways to summarize the information in the six standard metrics graphs, Figures 1 to 6. Graphic summaries of three features are especially helpful:

- The variability of the measure of interest
- A reference measure of central tendency
- The variability in relevant time intervals.

These features lead to another series of graphs, with the vertical axis showing the measure of interest and the horizontal axis indicating time. Different projects have different time intervals for system test and controlled introduction. At any point in time, different software releases have been available and used by customers different amounts of time. Each summary graph presents certain key values from the individual software releases. Each graph also contains a line indicating the average value of the measure of interest. Each summary graph thus shows the the variability of the individual software release values about the averages and over time. Examples are the six summary graphs selected for the semiannual Software Quality Assurance Report:

- *Cumulative Fault Density at Release—Faults Found Internally.* The value of the cumulative fault density at the time of release for each participating software release is plotted in this summary graph (Figure 7). The horizontal axis shows the time interval from the start of system test until release for each participating release. The horizontal positions highlight the variation in testing time, which is important in evaluating the fault density values. The vertical position for a given project—the value at release—is a key point from the project managers' and the customers' viewpoints.
- *Cumulative Fault Density—Faults Found By Customers.* This summary graph shows the cumulative values of fault density for faults found by customers at the graph date, for all software releases (Figure 8). The horizontal axis indicates the time since a given project has been released.
- *Total Serious Faults Found.* This summary graph shows the total number of serious faults found between the start of controlled introduction and the report date (Figure 9). This summary graph does not permit general comparisons across all projects because projects of different sizes are likely to have different total numbers of faults. But it does give an indication of the overall variation in this measure.
- *Mean Time to Close Serious Faults.* Because this summary graph shows the mean time to make corrections for each project as of the graph date, comparisons are valid. This graph (Figure 10) makes it easy to see how quickly, on average, serious software problems are being solved.
- *Mean Time Still Open for Serious Faults.* Because this summary graph (Figure 11) shows the mean time that serious faults on a project remain uncorrected, as of the graph date, it too is good for comparisons.
- *Total Field Fixes.* This summary graph (Figure 12) shows the total number of field fixes, as of the graph date, for each software release in the time since its controlled introduction was begun. Releases can not be

Figure 7. Cumulative fault density at release—faults found internally.

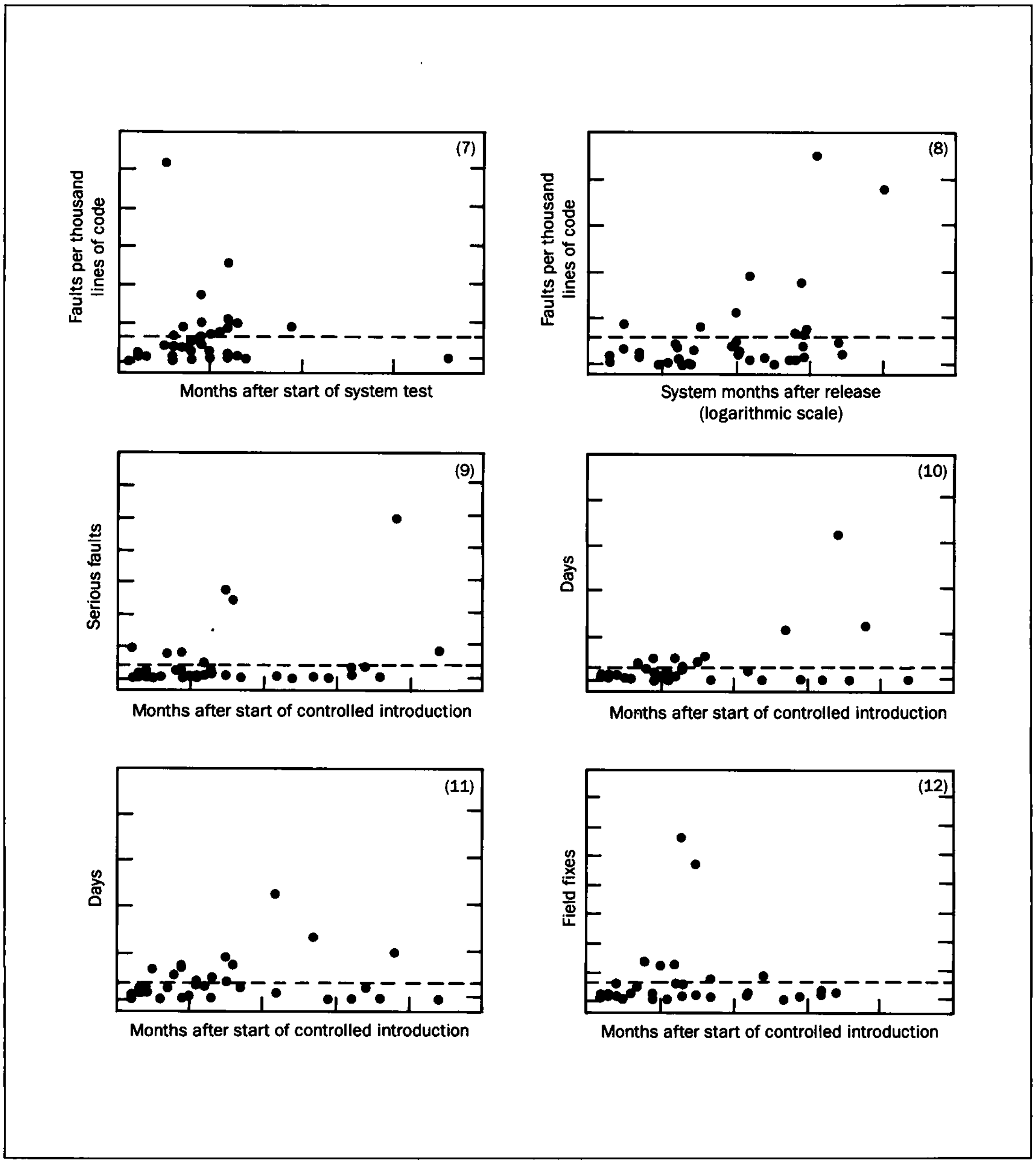
Figure 8. Cumulative fault density—faults found by customers.

Figure 9. Total serious faults found.

Figure 10. Mean time to close serious faults.

Figure 11. Mean time still open for serious faults.

Figure 12. Total field fixes.



compared directly on this graph because they vary in size, but the overall status and variability are evident.

Conclusion

The standard software quality metrics have been in place for several years. A system for collecting the data, training the project personnel on how to provide the data, and reviewing data submitted for correctness has been set up to ensure reliable, accurate reports. Most recently, 42 projects provided data on 73 releases. This system of measuring software quality is an important part of Bell Laboratories' continuing efforts to improve the quality of the software it develops.

(Manuscript received November 19, 1985)