Reducing software variance
by using a manufacturing process
to control software quality

Isaac Levendel

Because of the nature of human thought, any complex productive human activity is iterative in nature, a sort of conquest of the unknown. Generally, each iteration results in an improvement of the intended product, and sometimes, in a setback. Software manufacturing is no exception, and this will remain true until it becomes possible to separate software manufacturing from the programmer. The iterative nature of the software manufacturing process causes a blurring of the manufacturing process phases. Coupled with low testing budgets, this difficulty results in poor field releases: product improvement must extend far into the field deployment phase. A comprehensive model is proposed to pattern the software manufacturing process after the hardware manufacturing process, in spite of fundamental differences between the two processes. The failure data produced in each process phase is analyzed and used to control the movement to the next phase. At each phase, the error rate can be compared to quality criteria, and a decision can be made whether