
The Editor's Page

SOFTWARE DESIGN = ENGINEERING DESIGN?

A useful consequence of the recent hype over the "Y2K Bug" is recognition that a preponderance of today's engineered systems depend for their functioning on embedded software. From the humble electric toaster and dishwasher, to the latest airliners and sophisticated medical scanning devices, proper performance, whether reflected in unburned toast, clean dishes, or the safety of travelers and patients, demands that the operational software exhibit the highest degree of reliability.

Unfortunately, those of us who regularly use computers have good reason to question whether or not our confidence in current software systems may be misplaced. Frequent operating system crashes, unintelligible error messages that appear from time to time on our screens, new software versions containing more and more features most of us will never use but still lock up our keyboards, announcements of bug-fixing patches even before release of Version 1.0 of a new system, suggest that few in software development believe in, or perhaps have even heard of, six sigma.

Conversations with computer science educators do little to allay our concerns. They report that students seeking computer science degrees (a perceived ticket to a high-paying job) rebel at increased requirements in rigorous foundation areas like discrete mathematics and statistics—"just teach us to write code!" Some college administrators, seeking to increase enrollments in the popular computer science area, may actively encourage less rigorous curricula. Requirements that graduates demonstrate a commitment to professional and ethical responsibility, specified for accredited engineering programs by *Engineering Criteria 2000* of the Accreditation Board for Engineering and Technology (ABET), appear to be the rare exception in programs aimed at preparing computer professionals.

Even more disturbing are reports from industry that many experienced programmers resist efforts to impose a system of disciplined software development, careful documentation, and formal review of code before software testing begins. Can one conceive of designing a critical piece of hardware, for example, a braking system for an aircraft landing gear, without such requirements? Yet one wonders, if the antilock braking features depend on embedded software, how much discipline went into that critical part of its development?

Some may perceive this as a radical suggestion, but I firmly believe that the time has come for software development to become the engineering discipline that it should be, and that a program in software engineering be the expected preparation for professional software developers. This does not mean that software engineering students must take courses in thermodynamics and engineering mechanics. It does mean that they must have a firm grounding in the mathematical and statistical bases of information processing; that they recognize the necessity of disciplined design procedures, documentation, and review; and that they demonstrate a commitment to the professional and ethical ideals of the engineer. This

cannot occur without coordinated and committed action by computer science educators, professional societies, and the cooperating accrediting bodies, ABET and the Computing Sciences Accreditation Board (CSAB). Equally necessary is recognition by state engineering registration boards and legislative bodies that the design of certain software can affect public health and safety as much as, or even more than, design of a bridge. The registration boards, along with the National Council of Examiners for Engineering and Surveying (NCEES), must also recognize that not all engineering is based on the physical sciences and modify the Fundamentals of Engineering Examination accordingly.

Until these things come to pass, even this confirmed technophile will feel a bit queasy about that new car with the steering system that depends on software rather than a mechanical linkage; that highly exothermic chemical reaction with digital, rather than pneumatic, temperature control; or that next airplane ride.

—John W. Prados

NOTICE: E-MAIL TO THE EDITOR

Although the *Journal of Engineering Education* does not provide space for letters the editor in its printed issues, e-mail correspondence regarding the Journal's papers is encouraged; such letters should be addressed to the editor jprados@utk.edu and will appear at the Journal's World Wide Web site, <http://www.asee.org/publications/html/editor.htm>. Letters will normally be posted on receipt; however, the editor reserves the right, in his sole judgment, to reject letters on the grounds of excessive length or poor taste.

Copyright of Journal of Engineering Education is the property of ASEE and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.