Major changes in federal software policy urged

Galen Gruman, Soft News Editor

"Government policies on everything from budgeting to intellectual property rights have congealed over time in a manner almost perfectly designed to thwart the development of quality software," concluded a US congressional subcommittee's staff report on federal software procurement and development.

The report outlines the problems in software development familiar to the software-engineering community but not to federal officials. It also recommends several actions to Congress to improve federal acquisition and development.

The report's main recommendations are a new basis for software-procurement decisions, removal of the waterfall development model from federal standards, expanding the Defense Dept.-funded Software Engineering Institute's mission to include nondefense federal agencies or duplicating its functions in other agencies, adding system-hazard analysis to safety-critical device certification, and incentives to attract — and keep — skilled software engineers in government.

The report also said the government would do well to consider whether software engineering should be made a separate discipline from computer science in academia and whether software professionals should be certified or licensed.

The report, "Bugs in the Program," was published in late October by the US Government Printing Office. *IEEE Software* received an advance copy. The report was researched for two years and written by staff members at the House Committee on Science, Space, and Technology's Subcommittee on Investigation and Oversight.

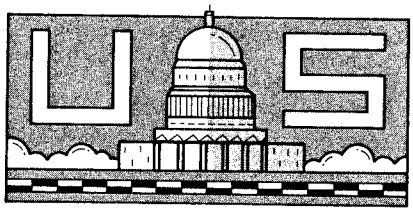
Problems masked by successes. "The fundamental intellectual foundation, even the appropriate mathematics," does not exist to solve the 20-year-old software "crisis," said William Wulf, the National Science Foundation's director of the Computer and Information Science and Engineering Directorate, at a Senate hearing on July 26. The report underscores this basic difficulty, concluding that "no short-term solutions to the problems identified ... are evident."

These problems have been masked by what the staff, in a transmittal letter accompanying the report, called software's "virtuous cycle." As new-generation machines add new capabilities, their prices go down. For the government, this offers the promise of gaining employee productivity through moderate expenditures.

But this virtuous cycle in hardware hides the opposite trend in software, the letter said. "As the complexity of systems increases, government managers find that the software they buy or develop does not achieve the capabilities contracted for, that it is not delivered at the time specified, and that the cost is significantly greater than anticipated," it said.

The report identified the following problems and recommended actions for Congress. The committee staff members are now working on legislation proposals that they hope to see introduced in the winter congressional session.

NEWS **FOCUS**



US Software Policy

Procurement. The "malign influences" of the federal budgeting system force federal software project managers to begin program development before they are ready to do so, which raises the overall cost. "The strange policy where the government pays twice for a system — once to buy it and once again to make it work the way it was expected — cannot be sustained in an era of multibillion-dollar shortfalls in the treasury," the report said.

The procurement system "does not take into account the special needs of ... software systems and can compromise effective software development from the start." It "frustrates" program managers by forcing them to write highly detailed specifications for systems whose boundaries are extremely fluid, by not letting them reflect what their users have learned, and making them move to production before all of the vital questions on how the system will ultimately perform have been answered, the transmittal letter said.

"What has become apparent is that system costs are driven not by hardware but by software," it said. In 1955, software development accounted for about 12 percent of a system, software maintenance 5 percent, and hardware 83 percent. In 1985, software development took about 40 percent, software maintenance 45 percent, and hardware 15 percent, the report said, quoting Barry Boehm's *Software Engineering Economics* (Prentice-Hall, 1981).

"While software now drives system requirements, the procurement system still focuses attention on hardware," the report said. It cited the Strategic Defense Initiative, "the largest and most complex software-development project devised by the federal government," as "a prime example" of this orientation. While SDI and Defense Science Board reports repeatedly stressed the criticality of software, SDI progress "is usually discussed in terms of hardware," the report said.

To address this problem, the report recommends that the National Aeronautics and Space Administration and the Office of Federal Procurement Policy "design new software-oriented procurement regulations reflecting today's reality in buying systems." The procurement office is trying to recodify federal procurement regulations into a single, greatly simplified statute applicable across all agencies.

Another problem is that procurement guidelines often forbid the use of evolutionary development models like Barry Boehm's spiral model, which advocate repeated testing of alternative designs throughout the process to determine which best meets user needs within resource constraints and acceptable risks. "These models reflect the commonsensical principle that any process can be improved with practice," the report said, but they conflict "with the realities of the present procurement system," which is based on the waterfall design model in which the system is fully defined before development begins.

Implementing the waterfall model requires a "legally binding contract specifying in excruciating detail exactly how the system will look at delivery some years hence," the report said, which means that changes made as development progresses "adversely affect software quality and increase development costs."

Recognizing the outdated reliance on the waterfall model, the revised DoD-Std-2167A *Defense System Software Development* standard that was adopted in February 1988 removed explicit references to the waterfall model. But the Defense Science Board Task Force on Military Software, which advocated the removal, warned that the revised standard "as a whole continues to reinforce exactly the document-driven, specify-then-build approach that we believe causes so many of [the Defense Dept.'s] software problems."

Procurement problems are worsened by the budgeting process, the report said, because "program costs are often underestimated at the outset" to get approval of at least some funds from a tight budget. "This hurts the program at the

crucial design and development stage and further complicates long-term planning," it said, because the costs of fixing errors during maintenance are typically six to 10 times more than fixing them during design. The costs for highly complex aerospace applications can be 100 times more expensive, it said.

Congressional pressure to minimize first-year costs (to at least get the program started) means a compressed, cost-driven schedule in which production decisions are often made before development and testing could find and fix systems errors, the report said.

As an example, the report cited a 1986 finding by the US comptroller general that the Federal Aviation Administration's air-traffic control system's development approach "does not adequately mitigate technical risks and does not provide for suitable operational simulation of the advanced automation features." Costs during the system's design-competition phase are already 50-percent higher than originally estimated, the report said, because insufficient initial funding required expensive fixes later on.

The report recommended that experiments between competing alternatives be funded up front and that prototypes be tested with real users "to determine whether the systems requirements reflect actual needs," it said. This means spending more money at the beginning to reduce the lifetime costs.

Changing budgeting tactics will not be easy, the report acknowledged: "No program manager relishes the thought of defending a request for funds when the major activity seems to be endless arguments over abstruse technical points by large numbers of well-paid engineers. Yet experience shows this is precisely the course to follow because it answers most, if not all, the questions that are expensive to fix on a production line."

Quality, reliability, safety, and security. Although stressing their importance, the report offered less-specific recommendations on how to achieve quality, reliability, safety, and security.

While acknowledging they are not universally accepted, the report described as a "proposed solution" the statistical quality-control methods originated by W. Edwards Deming for factories and advocated for software by Harlan Mills in his Cleanroom development approach. These methods use test cases that reflect statistical samples of user operations, not just selected test cases.

The report cited AT&T's reliability model, which is based on assessing the number of failures that occur during execution rather than the number of flaws in the code, as a "major contribution" to reliability assurance. (A failure is any departure of a program's operation from the program requirements.) The report noted that software with frequent failures cannot be trusted to perform mission-critical functions.

"Reliability is a necessary but insufficient condition to assure software safety," the report said. It urged that regulatory agencies not merely rely on reliability numbers, since some mishaps have occurred while the software was operating exactly as intended.

It cited the shooting down of an Iranian civilian airliner in July 1988 by the *USS Vincennes* as an example of how a system behaving normally endangered people's safety. The ship's Aegis warning system did not display the altitude of approaching aircraft in real time; had it done so, the ship's captain could have seen that the Iran Air plane was not descending toward his ship despite the reports he was getting from the antiaircraft-warfare station, the report said. The Aegis system is being reprogrammed to display such data.

The Defense Dept.'s Triservice Systems Safety Working Group has proposed a program to improve the government's ability to manage risks posed by software. It would establish an R&D center to research software-analysis and risk-assessment techniques and an applications lab to transfer the resulting tools and methods into practice. The Defense Dept. has not funded this effort, so the report recommends that Congress establish and fund a similar organization.

A General Accounting Office survey in May 1988 of nine system-development projects found that no agency treated information security as an integral functional requirement. The report recommended that the spiral development model be modified to include confidentiality, integrity, and availability attributes of security requirements during design, risk analysis, and development.

Assurance. Regulatory agencies have it even worse than procuring agencies, the transmittal letter said, because they must clearly understand all the assumptions underlying the systems. "Ideally, an agency would like to apply a standard analysis to a software-driven system and receive an unambiguous evaluation of its potential for creating hazards," the report said, "Such tools do not now exist." Statistical quality methods can help create such tools, the report said.

The report recommended that regulatory agencies use the Federal Aviation Administration's system-hazard analysis requirement in its aircraft-approval process as a model, since it seeks to identify the most critical failures and their causes and modify the design accordingly. The report noted that the FAA process does not — but should — explicitly consider

Recomendations in brief

The House science subcommittee's staff report, described fully in the accompanying story, makes these key recommendations:

- Establish a permanent software-policy apparatus, most likely under the White House's Office of Science and Technology Policy, to raise the profile of software issues and educate government agencies.
- Expand the Software Engineering Institute's role beyond the Defense Dept. or establish similar organizations for other federal agencies.
- Establish a group within the National Institute of Standards and Technology to evaluate development methods before they are introduced into federal agencies.
- Spend more money on development up front, including funds for experiments between competing alternatives, to save money over the project's life.
- Require hazard analysis, particularly for software, during the certification of safety-critical devices like medical systems.
- Eradicate all traces of reliance on the waterfall model of development from federal standards. Promote the use of evolutionary development models.
- Give program managers the flexibility to make decisions on software development as it progresses, rather than be bound by outdated detailed requirements.
- Restructure salaries to attract and keep software engineers in federal agencies.
- Focus research and management efforts on ensuring reliability, safety, security, and quality.

The report also suggested

- that the profession establish some certification, accreditation, or other basis on which customers like the government could evaluate a practitioner's software-engineering abilities, especially for safety- and relibility-critical systems, and
- that the government study whether software engineering should be a separate discipline from computer science.

software; FAA officials told the report's authors that they were considering adding software-specific analysis.

Personnel. Federal agencies increasingly need well-trained software specialists but cannot offer competitive salaries. Entry-level salaries for computer specialists were 22.7 to 31.6 percent less than in private industry, the report said. Experienced senior managers make 65 percent less than their industry counterparts.

NASA has reduced the difference in salaries by hiring most of its computer specialists as engineers, who are two salary grades higher, the report said. The report recommends new personnel policies at the National Institute of Standards and Technology and "a similar search for innovative hiring and salary structures at NASA." Other agencies will need such efforts, too, it said.

Many project managers have little flexibility to trade off technical performance, cost, and schedules; they can only make recommendations, the report said. Furthermore, most managers have no technical background, it said; their experience is in administration or finance.

Program managers are also expected to tailor development standards imposed by government contracts. "Fully imposing a complex standard like DoD-Std-2167A on every piece of software produced by the government would drive costs through the roof, would produce a mass of useless paper, and would reduce safety in critical systems by diverting management attention," the report said. Some managers have offloaded this burden by asking contractors how to implement the standards that apply to the contracts, the report said. Such contracts "should be monitored by agency managers to avoid serious trouble in the program," it said.

Other considerations — like budgets and politics — thwart technical management, the report said. For example, a contractor with little software experience may be chosen because it had the lowest bid, despite qualms about that contractor's abilities, it said.

Technology transfer. "Many federal agencies have already worked through the pitfalls of increasing complexity in software, but other agencies are not benefiting from that experience," it said. Because technology transfer "requires direct contact between technical professionals," the report urged the Office of Science and Technology Policy to oversee a program of long-term interagency coordination in software.

Education. Quoting a 1988 conversation with Nick Stewart, then president of the American Society for Quality Control, the report said the software-engineering field "still resembles a 'shoot from the hip' industry locked in an academic debate about its status as a discipline separate from computer science." Most students are "not taught an engineering environment, just coding," Stewart told the report's authors.

The report praised the NSF's reorganization that established a Computer and Information Science and Engineering Directorate, which gave computing stature equal to other fields. A 1988 Computer Society Board of Governors report said "undergraduate computer-science education in the United States has problems — serious ones — [that] must be fixed" and recommended increased funding for new teaching technologies, curricula, and lab equipment. The report acknowledged this report but said that the NSF's greatly increased budgets have addressed this need. "If computer-science education is not receiving sufficient support, this may be a problem with the mechanism for allocating directorate resources at NSF," the report said.

The failure of the software community to accept its responsibility for enforcing professional standards may lead to loss of prized autonomy, the report warned.

The report also cited several curricula-improvement efforts, including one at the University of Tennessee based on the Mills Cleanroom approach and the ongoing effort at the SEI.

However, the report acknowledged the Association of Computing Machinery's "core of computer science" task force report that recommended that software engineering should not be separate from computer science but be treated as a subarea. "On the other hand," the report said, "there may be some benefit to a more focused software-engineering curriculum, especially in supplying critical skills now woefully short in the federal government. A closer examination of this seems warranted."

Accreditation and licensing. "It is the right of government, when contracting for the purchase of software, to include any provisions that a prudent customer believes will assure a quality product," the report said, "If certification or licensing will achieve that goal, it is in the best interest of the government to require that contractors ... have these credentials."

Nancy Leveson, a software expert at the University of California at Irvine, said she might support limited certification, such as for safety-critical projects, as Britain now does, the report said.

After conducting hearings on integrity in scientific research, the subcommittee concluded that "a professional community should be permitted to enforce its own standards, so long as it demonstrates a fair, impartial, and expeditious consideration of these questions," the report said. But "failure of the software community to accept responsibility in this area may lead to the loss of prized autonomy," it said.

The report said the community was beginning to realize that, like it or not, it may have to accept some form of certification. It quoted ACM President Bryan Kocher's June statement that "it is time for ACM, with IEEE and others if possible, to propose and strive for adoption of appropriate federal standards for the computing profession."

Recommendations to Congress. In addition to its broad recommendations, the report outlined action items for the House Committee on Science, Space, and Technology to consider. Committee staff members are drafting legislation proposals based on these recommendations for possible introduction to Congress in early 1990:

- Form a working group on software-development improvement under the auspices of the Federal Coordinating Council on Science, Engineering, and Technology. (This council, based in the White House's Office of Science and Technology Policy, is made up of representatives from several agencies.) The report recommends that members of the working group come from the Defense Dept., NASA, FAA, NSF, NIST, National Security Agency, Internal Revenue Service, Health and Human Services Dept., and Social Security Administration. It would draw on government, industry, and academic contributions to design a research program to identify weaknesses in government software, the report said.
- The NIST's National Computer Systems Laboratory should develop the capability to analyze and evaluate proposed methods for improving government software development before they are used.
- The NSF should remain focused on basic research in next-generation software problems and in the improvement of software-engineering education.