

# Study criticizes Defense Dept. development, acquisition

*Galen Gruman, Assistant Editor*

A Defense Science Board task force on military software has recommended that the Defense Dept. move away from the waterfall model of program development and alter its software acquisition procedures. Technical developments will not increase dramatically in the next decade, it said, so planners should not rely on technical fixes to what are management problems.

The task force, chaired by University of North Carolina professor Frederick Brooks, Jr., said "today's major problems with military software development are not technical problems but management problems" and called for "a major reexamination and change of attitudes, policies, and practices concerning software acquisition."

Its major recommendations were that the Defense Dept.:

- Change the development method. "The most common present method of formulating specifications — issuing a request for proposal, accepting bids, and letting a contract for software delivery — is not in keeping with good modern practice and accounts for much of the mismatch between user needs and delivered function, cost, and schedule," the report said.

- Mandate the iterative setting of specifications, the rapid prototyping of specified systems, and incremental development. "DoD-Std-2167 needs a radical overhaul to reflect best modern practice. Draft DoD-Std 2167-A is a step, but it does not go nearly far enough," the report said. "DoD-Std-2167 should be further revised to remove any remaining dependence upon the assumptions of the waterfall model," it said.

"The problem if 2167-A is codified is that it will stay for a while" and that its important shortcomings will not be fixed for several years, said Barry Boehm, chief scientist at TRW Defense Systems Group's Office of Technology and a task force member, in an interview.

- Recognize that the military cannot compete for qualified software engineers with industry and thus concentrate its talent pool on acquisition instead of

construction.

- Use off-the-shelf, commercial software whenever possible. "The best way to save money in software development is not to develop," Brooks said in an interview. The Defense Dept. has moved in this direction, said Mary Shaw, a task force member and chief scientist at the Software Engineering Institute, in an interview. "The DoD is not a dominant player in the software area, so it makes a lot of sense to take advantage of what's going on in the market," she said. Unfortunately, "there's a tradition in the DoD that runs the other way," Shaw said.

- Put economic incentives in contracts to promote the reuse of existing

---

## ***The study says to remove dependence on the waterfall model and to focus the limited talent pool on acquisition.***

---

modules — "even in mission-critical embedded systems," the report said. Furthermore, the contracts should let contractors "profit from offering modules for reuse, even though built with DoD funds" and should "encourage contractors to buy modules and use them rather than building new ones."

Boehm recommended that contracts "fix the size of profit based on the quality of your product. There is necessarily an element of subjectivity there, but it seems to be a better way to get the incentives in the right directions" than traditional estimated-cost-plus contracts. These award-fee contracts would make it more likely that contractors will go beyond the specifications to create a better (and more profitable) product and that purchasers would be more willing to spend more money to get better quality, he said.

This also means that the policies on who gets the rights to software funded by the Defense Dept. but developed by

a contractor must be rethought. Two more-equitable rights proposals are being considered.

- Develop metrics for software quality and completeness and use those metrics routinely in contracts.

- Strengthen the mandates to use Ada. (Recent directives have done this, Brooks acknowledged.) Acquire the tools needed to use Ada and implement modern software-engineering practices.

- Forbid the creation of Ada subsets.
- Forbid use of fourth-generation languages unless using Ada or another general-purpose language would cost at least 10 times as much over the code's lifetime.

- Move the Software Technology for Adaptable, Reliable Systems Joint Program Office to the Air Force Electronic Systems Division under a flag-rank officer also responsible for the Software Engineering Institute and the Ada Joint Program Office, a move intended to raise the programs' visibility and improve cooperation among them.

Ronald Kerber, the deputy under-secretary of defense for research and advanced technology, opposes the move of STARS to the Air Force because he believes the program should not be too closely tied to any one service, Brooks said. (Kerber declined to comment, but a written statement confirmed his opposition to the move.) Brooks disagrees: "Keeping it [STARS] near the working people [in a service] will make it more transferable," he said.

The report acknowledged that many of its findings echoed those in earlier studies, but "most [earlier recommendations] remain unimplemented," it said. Defense Dept. spokeswoman Susan Hansen said several improvements suggested by the report have been made since the nine-member task force first convened in May 1985, but she also said that many of its major recommendations would require funding and legislative changes to be implemented.

Two areas the report does not address — the use of PCs and third-party software — will become major management issues, Brooks said.