Software Development Standards and the DoD Program Manager

Paul V. Shebalin

During the last half of this century, the Department of Defense (DoD) has made an enormous investment in computer-based systems. To control the cost, timeliness and quality of automated defense systems, DoD established a framework of military standards and specifications. A recent policy change (Perry, 1994) removed the requirement for DoD program managers to adhere to this framework; nonetheless, the necessity remains for applying effective contractual software development standards. This paper describes the purpose and intent of the current military standard (DOD-STD-2167A) dealing with software development, and presents a model of the contractual process required to implement the standard. It also outlines the process which has been used to update and issue software standards. It concludes that the proper application of any DoD software development standard will continue to be a difficult task which depends primarily on the capability of government program managers and which must accommodate the range of capabilities of individual software development contractors.

THE DOD SYSTEMS ACQUISITION FRAMEWORK

To help execute its assigned missions, the Department of Defense (DoD) acquires systems through a process of research and development, test and evaluation, and production. Many defense systems are automated; comput-

Mr. Shebalin is a Professor of Systems Acquisition Management at the Defense Systems Management College. He is a graduate of Old Dominion University and holds an MS in Computer Science from The George Washington University, an MBA from the University of Oklahoma, and an MCS from the University of Virginia.

ers and software are major components and provide the system in which they are embedded with increasingly sophisticated capabilities.

During the last 20 years, DoD has been increasingly criticized about its ability to manage the acquisition of automated defense systems. Currently, DoD is spending approximately 10 percent of its budget on software life-cycle costs, and that proportion is expected to increase. Three general problems identified with regard to the software acquired by DoD are: it is always late, it always costs much more than estimated, and it does not work as specified (Kitfield, 1989 & Richards, 1990).

To appreciate the factors involved with software development standardization, it is important to understand the DoD acquisition environment. Although all levels and organizations within DoD contribute to the acquisition of automated systems, the focus of activity is the contracting agency and, within that agency, the program management office (PMO). Headed by a program manager (PM), the PMO is the organization charged with acquiring a "new or improved materiel capability" (DoD, 1991) as part of carrying out a program of acquisition. That responsibility includes contracting with a software developer (or developers) to produce the necessary computer programs. The individual computer programs are referred to as Computer Software Configuration Items (CSCIs) (DoD, 1985). For a particular acquisition program, a PM typically will be required to contract for and acquire a number of CSCIs. Although these CSCIs may be completed and delivered at different times, collectively they comprise the "software" which is subject to the general problems identified above. At any time, the DoD software acquisition process involves hundreds of PMs, within many separate contracting agencies, managing their individual acquisition programs, and thousands of contractors developing software for defense systems.

An acquisition program is the basic framework within which a PM operates and within which standards are applied. As defined by DoD Instruction 5000.2, an acquisition program is carried out in five phases: concept exploration, demonstration and validation, engineering and manufacturing development, production and deployment, and operations and support. The activities with which a PM is concerned in each acquisition phase are described in a number of places (e.g., DoDI 5000.2) and will not be addressed in detail here. It is important to note, however, that the first four phases of the life cycle of a DoD acquisition program involve the development of defense system software, while the last phase (operations and support) involves both the maintenance and modification of that software and the development of new software.

Operations and support is a very important phase. Even 10 years ago, 70 percent of the typical defense system's life-cycle software cost was

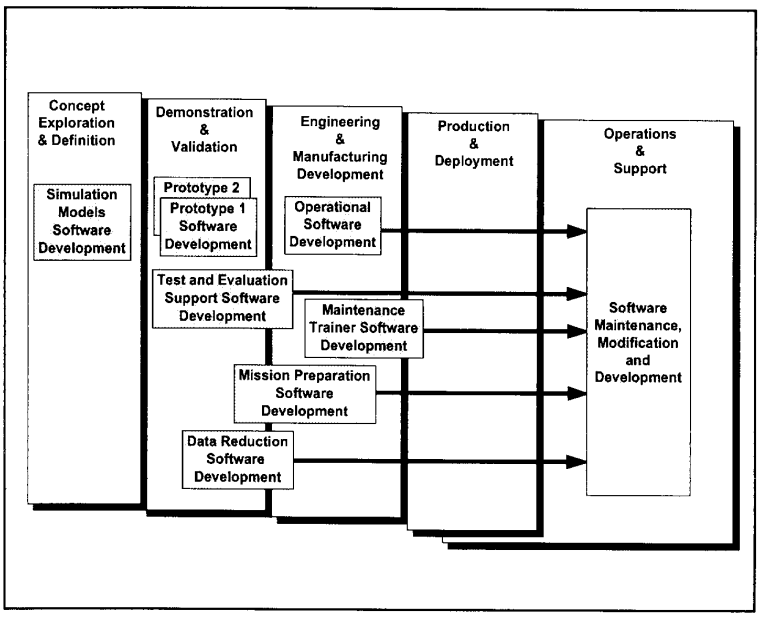

Figure 1. **Software and the DoD Acquisition Program Phases**

incurred during operations and support (Boehm, 1976). As depicted in Figure 1, the situation we have is one in which an enormous amount of software is developed during the formative period of a defense system and is maintained for 25 or 30 years. The software products, or CSCIs, in Figure 1 are provided only as an example since each acquisition program is unique in its software product requirements. For an acquisition program, a number of different CSCIs may be developed by several different contractors and then transferred to the care and maintenance of a single post-deployment software support activity. Obviously, the quality of the software and its documentation is a crucial factor in the ability of government agencies and support contractors to effectively maintain and enhance the software product.

The problems and opportunities created for DoD PMs by the use of automation and software technology will not go away on their own. The systems which PMs deliver depend more and more on computers and software; the current DoD defense acquisition strategy indicates that this will continue to be true for the foreseeable future (DoD, 1992). What role do the DoD software development standards play in helping—or hindering—the PM? This question is addressed below.

PURPOSE OF THE DOD SOFTWARE DEVELOPMENT STANDARDS

Software development falls under the larger purview of systems engineering. (Systems engineering will not be discussed here; for a good description, see the text by Eisner.) Within DoD system engineering, there are a number of interlocking and mutually supporting system development standards; software is only one area. It goes without saying that the integration of all of the DoD standards into a consistent, comprehensive set is a difficult, on-going task.

The intent of the DoD system development standardization has been to provide a common terminology, a uniform management process framework, an effective basis for educating DoD systems engineers and managers, and a stable, well-understood foundation for tasking the many contractors involved in DoD system development.

But, what is a standard? Words often have multiple, varied meanings and, when used to describe non-trivial concepts, especially in combination with other words, may lead different individuals to widely disparate conclusions about the fundamental concepts at issue. "Standard" may have one of several definitions (Webster's Dictionary), including "a criterion," "a model or example," "a rule for the measure of quantity, weight, extent, value, or quality," "a test of quality," and "any rule, principle, or measure established by authority." It seems reasonable to select the last definition as our starting point. Extending that definition leads us to capture the meaning of *DoD System Development Standards* as "*the rules, principles, and measures of system development established by the Department of Defense*."

Within DoD, such standards (technically referred to as *military standards (MILSTDs)*) are actually documents which establish rules, principles and measures for different aspects of system development, including engineering management (DoD, 1985), configuration management (DoD, 1992), software quality (DoD, 1988b), and software development and documentation (DoD, 1988a). While each of these areas of system development, and many more, are essential, we will only address the area of software development and documentation.

In the context of the acquisition framework described previously, then, the appropriate definition of *DoD* software development standards is:

> The documents approved by the Department of Defense which define the rules, principles, and measures which Program Managers apply during the acquisition, development, and support of software systems.

This may seem strange to some readers who might argue that the DoD software development standards are actually applied by software developers, not by government PMs. As described later in this article, this may have once been the case, but careful examination of the current DoD software development standard (DoD, 1988a) will support the accuracy of the definition provided above.

EVOLUTION: FROM MIL-STD-1679 TO DOD-STD-2167A

It may also seem strange that DoD software development standards are referred to in the plural: *standards* instead of *standard*. Why would DoD sanction the parallel use of more than one standard? The answer becomes obvious when we consider that the automated systems acquired and supported by DoD have a relatively long life, perhaps being deployed and operated for a period of 20 or 30 years. In the last 15 years there have been four distinct DoD software development standards:

- **MIL-STD-1679** MILITARY STANDARD: Weapon System Software Development, 1 December 1978
- **MIL-STD-1679A** MILITARY STANDARD: Software Development, 22 October 1983
- **DOD-STD-2167** MILITARY STANDARD: Defense System Software Development, 4 June 1985
- **DOD-STD-2167A** MILITARY STANDARD: Defense System Software Development, 25 February 1988

The effectivity of these MILSTDs has been sequential; that is, each new standard, on the date of issuance, has superseded the previous standard. But this only means that, as of the date of issue, PMs were required to use the new standard in establishing contracts with software developers. Developers with contracts already in place were obligated to continue performing under the provisions of their current contract, and that meant that any previously invoked software development standard continued to be in effect. Figure 2 shows this phenomenon: A particular software development standard remains in effect for 2 to 5 years, while the acquisition programs and their associated contracts continue until the affected systems are retired from service. The MIL-STD-SDD refers to a military standard, not yet issued, which will supersede DOD-STD-2167A.

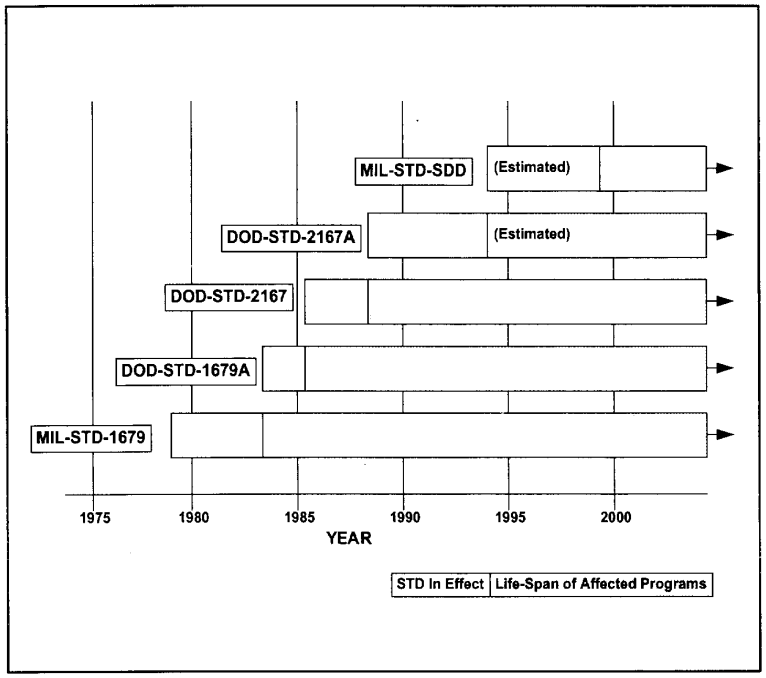

Figure 2. **Application and Effect of the DoD Software Development Standards**

The stated purpose of these standards has significantly changed as we have gone from MIL-STD-1679 to DOD-STD-2167A. The former was said to establish "uniform requirements for the development of weapon system software within the Department of Defense." It also stated that "Strict adherence to the provisions of this standard will ensure that the weapon system software so developed possesses the highest degree of reliability and maintainability feasible" (DoD, 1978). Unfortunately, the PM's understanding of "strict adherence" may have been nebulous, at best.

It seems that MIL-STD-1679 was often applied without proper interpretation by a government PM. This gave the software developer inadequate direction and, because of a narrow definition of the software development process, little room for innovation. The standard assumed the *waterfall* model of software development (Royce, 1970), and often put the government PM and the software development contractor in an adversarial position when the latter attempted to incorporate early prototyping or some other non-waterfall approach.

On the other hand, DOD-STD-2167A was written to allow the contractor more flexibility. I am aware that not every one will agree with this, but a thoughtful examination of DOD-STD-2167A will bear this out. As stated in the Foreword to DOD-STD-2167A, "This standard establishes uniform requirements for software development that are applicable throughout the system life cycle." This sounds fairly similar to MIL-STD-1679, so how has the contractor's flexibility changed? The answer is found further in the Foreword:

> This standard [DOD-STD-2167A] is not intended to specify or discourage the use of any particular software development method. The contractor is responsible for selecting software development methods (for example, rapid prototyping) that best support the achievement of contract requirements.

Also, DOD-STD-2167A specifically reads "this standard must be appropriately tailored by the program manager to ensure that only cost-effective requirements are cited in defense solicitations and contracts." The DOD-STD-2167A allows sufficient flexibility in software development and contracting. The difficult task, however, is not in understanding that the current DoD software development standard provides flexibility, but is in actually applying the standard as part of the contract solicitation, award and management process.

APPLICATION OF THE DOD SOFTWARE DEVELOPMENT STANDARD

The application of software development standards within DoD is not a simple, automatic process. In practice, due to several complicating factors, the application of these DoD software development standards has often been hit and miss. This is not necessarily an indictment of the standards; it is an observation of a situation which has arisen due to the constraints in time, funding, and personnel. These limitations notwithstanding, we present here and describe an ideal process of applying software development standards. This process model is intended to help both the DoD agency and the software development contractor to understand and better deal with the shared responsibility of developing high-quality automated systems.

A graphic representation of the application process for a DoD software development standard is provided as Figure 3. The primary organizations involved in carrying out the necessary activities are depicted as circles. The rectangles represent the documents which are intended to contain the information necessary to properly carry out the process.

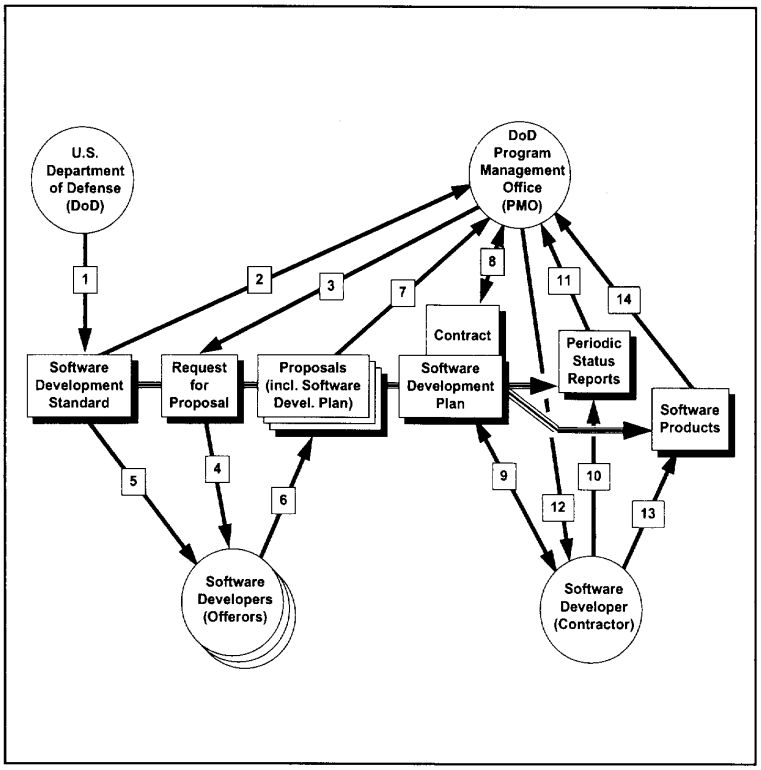

Figure 3. **The Process of Applying a DoD Software Development Standard**

Arcs from an organization to a document mean that the indicated organization is responsible for preparing that document. Arcs from a document to an organization represent the use of the information contained in the document. The heavy, three-part arrow underlying the documents represents the concept that each document must be developed on the foundations provided by the preceding documents. In this ideal process model, we will assume that the documents are complete in the information they should contain. In real life, these important documents are often grossly incomplete. Although the process is described in terms of DOD-STD-2167A, it is valid for subsequent DoD software development standards as well.

The arcs in Figure 3 are numbered; these numbers represent the sequence of steps taken in applying a software development standard to

a particular contract and producing a deliverable software product. Step 1 involves the DoD preparing and issuing a military standard (e.g., DOD-STD-2167A) for software development. As mentioned previously, the current standard is DOD-STD-2167A. As of the writing of this paper, MIL-STD-SDD (expected to be identified as MIL-STD-498 upon issuance), the follow-on to DOD-STD-2167A, is in final review. Alternatively, Step 1 may involve a nongovernment standardization organization, such as the American National Standards Institute (ANSI), issuing a commercial software development standard. Step 2 of the process requires the PMO to review, understand and incorporate the requirements of the software development standard into a contract. This incorporation has been termed tailoring and is a time-consuming, detailed process if it is done correctly. It is time-consuming and detailed because an individual, or individuals, must determine specifically which provisions of the standard must be required of a contractor and which provisions must be excluded. This is true for both military and commercial software development standards.

The provisions within DOD-STD-2167A indicate what is required of the software development *process* to be used by the contractor to develop the desired software product. That is, DOD-STD-2167A does not prescribe any particular process; it is up to the contractor to organize his software development process based on the provisions of the contract. This standard has played a central in providing DoD program managers a consistent, uniform basis from which to prepare a contract. This brings us to Step 3.

Once the PMO has adequately interpreted the requirements of DOD-STD-2167A and has decided on the software development requirements for their contract, what happens next? The answer is in a document called a Request for Proposal (RFP). The RFP is a solicitation for interested contractors to prepare and submit a proposal describing their approach to and understanding of the work required by the contract. Step 3 represents the preparation of an RFP by the DoD PMO and the release of that RFP to interested contractors. Details of the contents of an RFP will not be discussed here except to say that four sections of the RFP which are very critical to our process are the Statement of Work (SOW), the System Specification, the Contract Data Requirements List (CDRL) and the Instructions to Offerors. The SOW defines the tasks to be performed by the contractor, including the software development tasks required by and invoked from the software development standard. The System Specification specifies the desired characteristics of the system to be developed, including characteristics of the software product. The CDRL specifies the documents to be delivered under the contract,

including the software documents defined by the software development standard. Finally, the Instructions to Offerors section of the RFP gives the contractor directions on how to prepare and submit a proposal; for our process to work, the Instructions to Offerors must require the offeror to submit, as part of the proposal, a Software Development Plan (SDP). In summary, Step 3 represents the translation, by the PMO, of the *general* software development process requirements in DOD-STD-2167A into software development process requirements *specific* to the automated system whose development is to be contracted out. These specific requirements are contained in the RFP.

After the RFP has been released, a number of contractors will obtain copies, review the document and decide on whether or not to submit a proposal and compete for the contract award. This is shown as Step 4. At this point, a contractor will hold the primary printed document identifying the contract software development requirements—the RFP. Because the RFP may refer to many of the specific requirements in DOD-STD-2167A, rather than repeat them verbatim, the contractor may need to review that Military Standard. This is depicted as Step 5. After reviewing the RFP and DOD-STD-2167A, and deciding to prepare and submit a proposal, the interested offerors do just that, and, as represented by Step 6, deliver to the DoD their proposals and preliminary SDPs.

At this point, the application of DOD-STD-2167A is essentially complete. It is now up to the contracting agency (PMO) or, what is officially called a Source Selection Authority, to review the various proposals and select a contractor; this is depicted by Step 7. We do not expect the SDPs submitted by separate, competing contractors to be similar and, in practice, they are often quite different. The software development process model defined in each of these plans may also be very different. Each of these process models may be a reasonable and adequate interpretation of the contract requirements and may comply fully with DOD-STD-2167A.

The rest of the process is straightforward. The contracting agency (PMO) selects one offeror and (Step 8) negotiates with and awards the contract to that offeror. The SDP submitted by that offeror becomes part of the contract; the contractor is obligated to conduct software development as defined by the SDP (Step 9). This does not mean that the contractor (and the PMO) is stuck with a rigid, inflexible plan. Quite the contrary: as the contract is performed and the need to change the SDP is evident, the contractor prepares and submits status reports (Step 10) to the PMO. These status reports may contain recommendations that the software development process, schedule or other feature be

changed to adjust to emergent requirements. The PMO considers these recommendations (Step 11) and provides direction to the contractor (Step 12). The contractor modifies the SDP accordingly and proceeds with developing the software products. Ultimately, the process being utilized by the contractor produces a software product (Step 13) which is evaluated and accepted by the PMO (Step 14).

In summary, the application of a software development standard will result in a *plan* and a *process*. Ideally, by following the plan and adhering to the agreed-to process, the contractor develops software in a controlled, well-engineered fashion, the contracting agency understands and is able to track development progress, and the resulting software and documentation are of high quality. The effectiveness of the software development standard is dependent on the content of the contract clauses, the SOW, the CDRL, and the contractor's software development process.

EVOLUTION OF THE DOD SOFTWARE DEVELOPMENT PROCESS

The process defined above takes as one of its axioms that the role of the DoD software development standard is to provide 1) guidelines to the contracting agency for specifying the required contractor software development activities and 2) the basis with which the contractor can interpret the contracting agency's requirements in developing a responsive SDP. The current DoD software development standard, DOD-STD-216A, is a critical document providing the foundation for all of the DoD automated systems acquired during its effective period. But what if some of its provisions are less than optimal for procuring quality software products, either because of some inherent difficulties in the software development standard or because technology has advanced to the point that the standard's provisions are inconsistent with modern programming practices and techniques?

There is no doubt that technology will change and it would be overly optimistic to believe that any document, DoD or otherwise, could be written in a flawless manner. The DoD and commercial software development standards have been changed in the past and will continue to be changed. Is the modification and issuance of a new software development standard a fool-proof, efficient process? As with any group activity, the answer is an obvious *no*. But, is DoD's process for updating its software development standard reasonable and effective in meeting the demands of its users? We believe the answer is yes, although the process is certainly not perfect; no human activity is.

Since MIL-STD-1679, there have been three new software development standards. The process depicted in Figure 4 has been, in general,

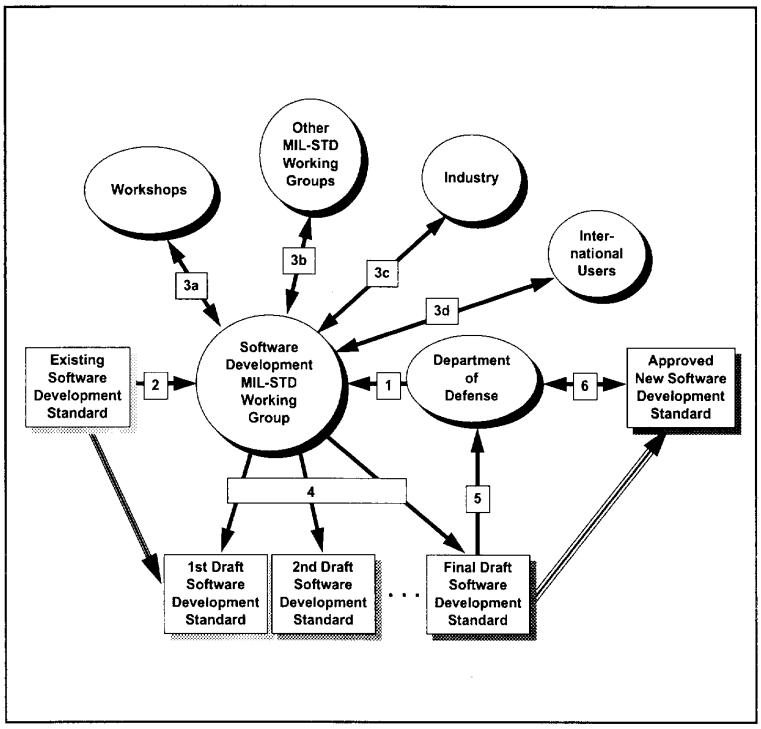

Figure 4. **The Process of Developing a New DoD Software Development Standard**

the process used to develop each of these standards. Because the DoD software development standards play such a central role in the process of system and software acquisition, it is important that the users be allowed to contribute to their evolution. The process of evolution we are going to describe is the process by which the standards after MIL-STD-1679 have come into existence. It is the process by which the pending standard MIL-STD-SDD is being formulated.

As shown by Figure 4, there are six basic steps used by DoD in the development of a follow-on standard to an existing military standard for software development. The first step involves the establishment and chartering of a working group by DoD. In the case of DOD-STD-2167, this was done in 1978 by the Joint Logistics Commanders (JLC) who established a Joint Policy Coordinating Group for Computer Resources Management (JPCG-CRM) and an associated Computer Software Manage-

ment (CSM) Subgroup to develop a follow-on to MIL-STD-1679. The CSM Subgroup was the working group which coordinated all of the activities necessary to develop DOD-STD-2167. In the case of the pending MIL-STD-SDD, the JLC, through the JPCG-CRM, established the Harmonization Working Group (HWG) to develop the DoD's new software development standard; that development and coordination is currently in progress. While members of these working groups are primarily DoD and other U.S. government employees, there may be one or more participants from private industry.

In Step 2 of the process, the working group examines the current DoD software development standard, reviews the pertinent criticisms, and develops a specific organization and plan for determining the changes necessary to transform the existing military standard into a new one. The primary focus of the plan is to identify the type, number and schedule of activities that will be used to involve the various interested users in the development of the new standard. Step 3, obtaining comments, suggestions and criticisms from interested parties, and Step 4, preparing the working draft documents, of the process are the longest and represent the majority of the effort. These steps are managed as parallel sets of activities by the working group. In Step 3, the working group establishes relationships with several constituent groups for the purpose of generating discussion on desired modifications to the current software development standard. A workshop (3a) is one type of forum that has been used to great advantage by the DoD working group. Workshops are held once or twice during the deliberations on a new standard. For the purpose of obtaining timely input from the DoD software development community during the development of DOD-STD-2167, JLC Software Workshops were held at the U.S. Naval Postgraduate School in 1979 and 1981, with 80 and 100 participants respectively. In preparing for MIL-STD-SDD, similar workshops were held in San Antonio, Texas. Additionally, the software development standard working group solicits comments and suggestions from other military standard working groups (3b), industry groups (3c) such as the Council of Defense and Space Industries Association (CODSIA) and international users (3d) such as the German and United Kingdom Ministries of Defense.

This coordination and information gathering continues for a period of 2-3 years. During this time, working drafts of the new software development standard are published and distributed for review and comment to the working group members. The sifting and incorporation of comments is performed with the services of a support contractor to the working group. For MIL-STD-498 that support contractor is Logicon, Inc., of San Diego, California. When the working group is satisfied, a

final draft of the software development standard is published and submitted to DoD for review and approval (Step 5). In the final stage, Step 6, the new software development standard is approved and issued by the DoD as a military standard. At this point, the working group has fulfilled its charter and done its job; the working group is dissolved and its members and supporting agencies are released from their obligations.

OBSERVATIONS

The DoD software development standards are fundamentally different from the "commercial" standards used in industry for products and services. These software development standards are used as part of the contractual process by which DoD initiates the development of automated systems. The program manager determines contractor tasking, in part, by using the process requirements specified in a software development standard. Without such a document to draw from, the PM is left to uniquely determine software development terminology, documentation and tasks. The common use of one software development standard goes a long way towards ensuring that individuals, both government and contractor, can transfer from one automated system development program to another without a great deal of retraining. Similarly, a contractor will be less likely forced to change an established internal process to accommodate new terminology, new documents and new task definitions. Program managers, and the contractors who support them, have a difficult enough job developing DoD software without having to deal with a new software development paradigm for each separate program.

Even in the current climate of change and preference for "commercial" standards, the following conclusions can be made:

- No commercial standard exists which could replace DOD-STD-2167A (or the pending MIL-STD-498).

- The evolution of the DoD software development standards will continue. Changes in technology, differences between acquisition programs, and other factors will keep pressure on DoD to adapt. The adaptation process has a cycle of several years.

- Whether the standards are developed by working groups within DoD or by industry-based groups, the basic process of application described above will remain the same.

- The program manager cannot escape the responsibility of deciding which provisions of a software development standard to place on

contract. By their nature, all general software development standards used as part of the contractual process, as described above, will require interpretation.

- The proper application of any DoD software development standard will remain a difficult task. The particular nature of any program requires that such a standard be tailored and the appropriate provisions incorporated into the contract, either directly or by reference.

- Training and education of PMO personnel will continue to be a key ingredient in managing a process which Brooks described as a "monster of missed schedules, blown budgets, and flawed products" [17]. The preparation of contractual direction, starting with the RFP, must be effectively carried out if there is going to be any significant progress made in improving DoD's management of software acquisition.

The DoD software development standards have been and will continue to be necessary. The issue is not that a particular software development approach or process must be used by a contractor, but that *some* effective approach must be used. If this does not happen, then how can we expect the quality of automated defense systems to improve? The DoD software development standards exist to serve this end; they are the basis for determining the requirements which a contractor's internal software development process must meet. Standards such as DOD-STD-2167A help the program manager establish the minimum requirements for a contractor. These standards will continue to be an essential factor in defense systems acquisition, but their effect will only be as good as their interpretation and application by the PMO.

REFERENCES

Boehm, B. (1976, December). Software Engineering. *IEEE Transactions on Software Engineering,* Vol. C-25, No. 12

Brooks, F. P. (1987, April). No Silver Bullet: Essence and Accidents of Software Engineering. *Computer,* Vol. 20, No. 4. IEEE.

Defense Systems Management College. (1990). *Mission Critical Computer Resources Management Guide.* Fort Belvoir, VA: DSMC Press.

Department of Defense. (1991, February). Defense Acquisition. *DoD Directive 5000.1.* Washington, D.C: Department of Defense.

Department of Defense. (1991, February). Defense Acquisition Management Policies and Procedures. *DoD Instruction 5000.2.* Washington, D. C. Department of Defense.

Department of Defense. (1992, May). OUSD(A) Memorandum. Washington, D.C:

Department of Defense. (1988a, February). Military Standard: Defense System Software Development. *DOD-STD-2167A.* Washington, D.C: Department of Defense.

Department of Defense. (1988b, April). Military Standard: Defense System Software Quality Program. *DOD-STD-2168.* Washington, D.C: Department of Defense.

Department of Defense. (1985, June). Military Standard: Specification Practices. *MIL-STD-490A.* Washington, D.C.: Department of Defense.

Department of Defense. (1985, June). Military Standard: Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs. *MIL-STD-483A.* Washington, D.C: Department of Defense.

Department of Defense. (1978, December). Military Standard: Weapon System Software Development. *MIL-STD-1679.* Washington, D.C: Department of Defense.

Department of Defense. (1992, October). Military Standard: Software Development and Documentation. *MIL-STD-SDD (draft).* Washington, D.C: Department of Defense.

Eisner, H. (1988). *Computer-Aided Systems Engineering.* Englewood Cliffs, NJ: Prentice Hall.

Kitfield, J. (1989, July). Is Software DoD's Achilles' Hill? *Military Forum.*

Perry, W. J. (1994, June). Specifications and Standards – A New Way of Doing Business. *OSD Memorandum*. Washington, D.C.: Department of Defense.

Richards, E. (1990, December). Pentagon Finds High-Tech Projects to Manage. *The Washington Post.* Washington, D.C.

Royce, W. (1970). Managing the Development of Large Software Systems. *Proceedings of IEEE WESCON.*

Websters Ninth New Collegiate Dictionary. (1985). Springfield, Mass: Miriam-Webster, Inc.