

Management Understanding of Ada Sociology Necessary

by Donald G. Firesmith
Special to GCN

As Machiavelli noted, "There is nothing more difficult to take in hand, more perilous to conduct, or more uncertain in its success, than to take the lead in the introduction of a new order of things." Ada is such a technological transition, and the success of Ada projects often depends as much (or more) on the software manager's grasp and control of social, psychological, and political forces as it does on the manager's understanding of the important technical issues.

The software manager must be able to successfully deal with upper management in order to obtain the necessary funding for training, tools, and hardware; schedule modifications due to new life-cycles and the new emphasis on requirements analysis and design; and support for (and understanding of) the innovative approaches associated with Ada. Similarly, the manager must be able to negotiate with and educate contracting agency, user, and Independent Verification and Validation (IV&V) personnel. Finally, the manager must also lead an often diverse group of developers using new software development methods and tools in addition to a new language.

Although upper management Ada savvy and support is critical to the success of each corporation's Ada technology transition (as well as the success of its individual Ada projects), both corporate and project management typically exhibit a very disturbing lack of awareness of the important impacts of Ada. Ada and Ada-oriented software development methods are new and often looked on with suspicion by both technical and administrative managers whose experience is suddenly less relevant. The Ada transition is as significant as the one from vacuum tubes to solid state, and managers should clearly point out to upper management the dangers of hesitating (or resisting) during

such a technology transition, especially when the customer is demanding the change. The manager should also mention the advances and investments made by the competition.

Because upper management is often more oriented towards and receptive to financial arguments, the software manager should use Ada cost models such as SOFTCOST-ADA (tm) and provide a cost/benefit analysis. Practical information should be presented in terms of market and projected market share, financial impact, capital investment, personnel needs, corporate way of developing software, and the risks and means of managing those risks, etc. Above all else, Ada should not be oversold. Because of past over-optimistic promises prior to the recent proliferation of production quality compilers, initial projections of increased productivity may well need to be artificially low (e.g., 25%) to be believable. If rework is currently a major source of income (one that will decrease as Ada increases the quality of delivered code), the manager may need to show the projected impact of the current trend towards warranties and fixed-price contracts. The manager must also modify upper management's expectations regarding project phasing, milestones, and manpower loading, etc.

Finally, because people often are as motivated by personal and emotional reasons as they are by technical arguments, the manager should determine and discuss personal benefits with individual executives and how they may mitigate Ada's risks to their personal careers.

Contracting agency, user, and IV&V contractor understanding and support for Ada is also critical to project success. Yet contracting agency, user, and IV&V contractor personnel typically exhibit an even more disturbing lack of awareness of important Ada impacts than upper management. As Dr. Yale Jay Lubkin, Electronic Warfare Editor of "Defense Science and Electronics" noted, "With rare exceptions, the kind of people who procure and administer contracts are not the kind of people who can invent new ways of doing things, or even people who can understand and

appreciate a new way of doing things." They also tend to look on Ada and new Ada-oriented software development methods with suspicion, and this can be devastating during both the proposal effort and the formal reviews. Because IV&V personnel gain much of their expertise during the review of past projects, the IV&V contractor will tend to lag (like an anchor behind a ship) behind the state-of-the-art contractor during a technology transition.

In order to facilitate productive formal reviews, the manager should offer free training to contracting agency, relevant user, and IV&V personnel, and also work to convince the contracting agency to make this training mandatory on their IV&V contractor. The manager should work closely with the contracting agency, and ensure that he and his developers are prepared. The manager should have decided on the project software development method by the time prior to the time the proposal is written, and be aware of all relevant policy and mandates (e.g., DODD 3405.2 "Use of Ada in Weapon Systems") on the contracting agency. This will help the manager know how the proposal will be judged and what to do if he wants to use Ada and the Request for Proposal (RFP) does not mandate it. The manager should also know all relevant software development standards (e.g., DOD-STD-2167A "Defense System Software Development"), and how they should be tailored for Ada. He should work with the contracting agency to get them tailored to suit project needs; because different Ada-oriented software development methods require different tailoring, no one standard tailoring is suitable for all Ada projects. When necessary, the manager should justify requests for tailoring or deviations with the Acquisition Streamlining Directive (DODD 5000.43) which prohibits the government from imposing "how-to-manage" and "how-to-design" requirements on the contractor. The manager can also use recommendations of industry associations (e.g., CODSIA) and professional societies (e.g., SIGAda Software Development Standards and Ada Working Group). In order to

educate the contracting agency and ensure that IV&V personnel do not successfully raise improper complaints against your Ada-oriented development method during formal reviews, boilerplate should be avoided and the Software Development Plan and Software Standards and Procedures Manual should be complete and specific. The manager must ensure that the contracting agency understands the benefits of the proposed innovation and the need to modify their expectations regarding project phasing, milestones, etc.

When dealing with developers, the manager should be a leader (to new technical frontiers) and not merely an administrator. The manager needs to realize that resistance to Ada-specific ways of developing software is often due to ignorance and is a good clue to the lack of training, experience, and aptitude of some developers. To combat this, both managers and developers must be adequately trained (both scope and depth). Failure of training results in the failure to properly exploit many of Ada's most important capabilities and in using certain language features inappropriately because they are similar to features in older languages. This, in turn, results in the loss of many of the benefits promised by Ada. Managers should be aware that the two most common causes of trainee failure are (1) long experience with a single older language (especially assembly or C) coupled with an inflexible attitude and (2) negative management attitudes towards modern software engineering principles (e.g., "ivory tower garbage that keeps my developers from coding"), Ada (e.g., "just another oversold language"), and Ada-oriented software development methods (e.g., "bureaucratic waste of time"). Objective tests (e.g., the Ada Proficiency Test by Psychometrics) are available to test competency, both after training courses and during hiring.

Even if the manager and developers already have some Ada experience, it is often cost-effective to seek expert advice and use the experience of others to develop the project-specific Ada-oriented methods, standards,

and procedures. These methods, standards, and procedures should be publicly imposed so that the developers will clearly know what is expected of them. The manager should make it obvious to the developers that he or she supports and will enforce the project methods, standards, and procedures. Once a technical disagreement has been raised, discussed, and settled, the manager should enforce it and make it clear to the developers that the methods, standards, and procedures are subject to CONTROLLED change, but may NOT be ignored. Managers should ensure that their subordinate technical managers are TECHNICALLY competent and will enforce their technical direction. Managers should keep track of their developers and use Quality Assurance as their watchdog to identify the reactionary individuals who never make the transition to Ada, the lone-wolf hackers, and developers with the programming-in-the-small, edit-compile-debug mentality who will undermine management's new direction. Quality Assurance should also be used to find any developers using Ada inappropriately or using obsolete methods and backfilling documentation to "comply" with Ada-oriented project standards and procedures. They should also watch out for new (inexperienced) developers who have been taught modern Ada methods and are then being confused or intimidated by more experienced, but conservative developers. Each project will probably have at least one die-hard who will always resist the Ada approaches and who must be removed or isolated for the good of the project.

Managers must promote objective and egoless reporting because with Ada, they will have an even greater need for timely visibility into how the project is actually implementing the new Ada approach. Managers can ensure good communication among developers and between groups by requesting staff input, holding regular meetings to let developers be heard, and setting up a project newsletter or bulletin board to spread success stories, keep developers informed and provided with the big

picture, and ensure that lessons learned are captured and published.

Finally, managers must build team spirit, encourage cooperation, and remember what Ben Franklin said, "To get the bad customs of a country changed and the new ones, though better, introduced, it is necessary to first remove the prejudices of the people, enlighten their ignorance, and convince them that their interests will be promoted by the proposed changes; and this is not the work of a day."

Donald G. Firesmith is the software methodologist for Tactical Systems Division of Magnavox Electronic Systems Company, Chairman of the SIGAda Software Development Standards and Ada Working Group, and presents public seminars on Ada Project Management. (219) 429-4327