At the national and international level. The need for software standards, which seems straightforward at the project and multiproject levels, is less clear across company boundaries at the national and international levels.

The most obvious need is to be able to compare software development practices among companies. Today, any and all comparisons are, at best, vague. For example, the present databases on software fault discovery rates do not use the same definition for a fault. Some programmers call any problem a fault and fill out failure report forms for inclusion in the databases. Other programmers simply make corrections and include. Therefore, when two companies report software fault find rates of X faults per 1000 lines of code, they may or may not be saying the same thing.

The buyer's need to compare products is another reason for applying standards to the software development process, and a seller's adherence to IEEE/ANSI standards is a favorable comment on his product. In fact, something like the Underwriter's Laboratory seal on electrical appliances would be of tremendous value for software products, if the approval process were accepted by a large group of professionals.

The reasons for standards dealing with the software development process that we've noted are usually sufficient to start most organizations developing a set of standards. Then the questions really begin: "How will the standard affect creativity?" "What and when do we standardize?" Resolving these and other questions will require extensive discussion and effort.

Standards vs. creativity

The argument that standards restrict creativity is often heard in organizations that are dealing with new standards. Any conclusion reached is normally based upon the particular standard involved.

Music is a good example of a creative medium that has a rigorously standardized language. There are only so many notes, lines, and spaces. The mundane, simplistic features of documenting music have been thoroughly standardized, yet the creative arrangements of the notes are seemingly inexhaustible.

If software standards were applied in the same manner as music standards, the impact on creativity would be positive not negative. The creative effort in software development would be channeled toward the problems to be solved, not into areas that were standardized for that project. We could spend less time discussing what the document looks like and more time putting information in the document.

Document formats and descriptions are now typical standards, and symbols and/or syntax for describing a system's design or code are prime candidates for standardization. Such software standards, applied as suggested, would not restrict creativity.

The counter argument is that people will take a standard as an absolute and that this perception will stop, or at least slow down, the evolution of the technology area being standardized.

The opposing positions may be stated as follows:
- Standardization too early in the life of a technology may reduce its rate of development.
- Standardization too late in the life of a technology will delay the benefits of standardization.

The questions of when to standardize and what to standardize are closely related and should be answered at the national level by a consensus process. A single position selected by one author, by a small, homogeneous group, or by one company would simply reflect a limited set of conditions, not the national need.

The recent standardization of the Ada language and the ongoing effort to standardize a program design language for Ada are two good examples of software standards where debates over when and what to standardize are being resolved by a consensus process. Some opponents claim it is too early to standardize, even though Ada has been a Department of Defense standard for over a year.

In an organization, the decision to standardize may belong to one person, or, perhaps, to a committee. In these cases the questions become "Which standard?" and "How do we determine whether this is a reasonable standard?" A company can best resolve these questions if national-level, consensus-based standards are available as starting points. However, that raises specific questions about national-level standards within each development organization.

Providing the answers, directions to the answers, and a forum for ongoing pro and con discussions for software standards is the purpose of this department in IEEE Software. We look forward to the challenge.