In earlier Standards Departments, we discussed the IEEE software standards development process and some of the standards themselves. This kind of column and articles written by members of the standards development working groups for publication in various trade journals provide an important means of introducing the professional community to our standards.

Over the years, as people became aware of the IEEE standards and the benefits of using them, they requested more and more information from the IEEE. For those professionals who became really interested in certain standards and wanted to implement them in their own organizations, articles and copies of the standards did not provide enough detailed information for them to proceed.

In 1979, the first IEEE seminar for software standards was offered to the public. It was meant to provide the additional information people were then seeking about the first IEEE software standard on software quality assurance. The seminar was led by Fletcher J. Buckley, who had chaired the development of this first software standard. Since 1979, the seminars have grown and evolved to become the most effective way of disseminating detailed, practical information about the standards.

Realizing that thousands of people would eventually attend these seminars, the IEEE established a set of rigorous guidelines to govern the development and delivery of these presentations. Under these policies the IEEE can assure the quality of three major elements of the seminars:

- **Subject matter.** The subject matter of a seminar is kept current, practical, and usable by bounding or limiting the subject in three ways.
  
  First, each seminar is limited by the subject of the standard it covers. This may sound simple, but consider, for example, that there are now or will be shortly standards and seminars on the following subjects: software quality assurance plans, software verification and validation plans, software reviews and inspections, and test documentation.
  
  Overlapping is held to a minimum by carefully limiting the content of each seminar. This is particularly meaningful to the seminar participant who wants to learn how to perform a specific job now and does not want a broad brush covering several subjects. A result of this focusing is that each seminar is a standalone, usable package, and the entire group of IEEE seminars becomes an integrated set of training modules.

---

**The purpose of the IEEE standards seminars is to transfer technology from the state of the art to common practice.**

---

Second, the subject matter of each seminar is limited to software technologies that are usable today. Only certain technologies are usable right now. Applying the definitions of the life cycle phases of a technology provided in the accompanying box enables the seminar developers to select the technologies to be included in the seminars.

They do not want to spend classroom time on technologies that are too new. If a technology is still in the research laboratory and does not have a history of successful use in software development environments, it will not be included.

Neither do they want to include technologies that are old and familiar to a large percentage of the software community. If a technology appears on surveys of current practices showing a high use index, it is not suitable for these seminars. That is not to say that research and common practice technologies are ignored completely. They simply are not the focus of discussion.

The purpose of the IEEE standards seminars is to transfer technology from the state of the art to common practice. As a result of limiting subject matter in this way, updates must be made to the seminar materials as the state of the art progresses. Usually this means there will be an update about every 18 months.

Third, the subject matter is bounded by a consensus process. In developing the standard itself, as many as 200 or 300 people may have input to the standard. These individuals often work for two or three years to produce a written document that each of them can support. This long, difficult process of resolving terminology differences and working out agreements on technical positions results in what the IEEE calls a consensus-based standard.

When it's time for seminar development on a standard, the consensus process is put into play again, this time with only three or four people (developers/instructors) involved. Ideally, these individuals come from different parts of the industry working for different companies located in various areas of the country. The materials they develop will come from their various perspectives revised to a position agreeable to all of them. These consensus process materials reflect more experiences and insights than similar materials developed by any one person.

Also, the subject matter is limited by a consensus-based terminology (IEEE Std 729). If the definitions in the subject matter deviate from that established terminology, new definitions have to be noted for seminar participants. The most important outcome of using this consensus method to limit subject matter is a consistent, well-organized set of seminars.

---

**Instructors.** An instructor can make or break a seminar—even one that has well-
The life cycle phases of a software technology

The following is a characterization of life cycle phases of a software technology.

Research:
- Money is spent to create the technology.
- Applications using the technology are nonexistent.
- Methods and procedures for applying the technology are nonexistent.
- Standards and metrics are nonexistent.
- Tools are very limited and are custom-developed in the research laboratories.
- Training is available only in a few seminars and represents only the research studies.

State of the art:
- Money is spent to implement the technology.
- Applications using the technology are few and specialized.
- Methods and procedures for applying the technology are available from a few private sources.
- Standards and metrics are developed.
- Tools are limited and available from only a few private sources.
- Training is available in professional seminars.

Common practice:
- Money is spent to apply the technology.
- Applications using the technology are many and general.
- Methods and procedures for applying the technology are available from public resources and in textbooks.
- Standards and metrics are established and refined.
- Tools are readily available and in general use.
- Training is available in seminars and becomes a regular part of the university curricula.

Replacement:
- Money is spent to support the technology.
- Applications using the technology are complacent and met by a newer technology.
- Methods and procedures for applying the technology are static.
- Standards and metrics are well established and are used for economic research into automation of present methods.
- Tools are replaced by newer tools providing a higher level of automation.
- Training is limited to on-the-job efforts and is part of the university curricula.

Obsolete:
- Money is spent to eliminate the technology.
- Applications using the technology are nonexistent.
- Methods and procedures for applying the technology are falling into disuse.
- Standards and metrics are no longer noticed.
- Tools are finally stable, reliable, and unsupported.
- Training is not considered.

The only people who have firsthand knowledge of the arguments and agreements are the individuals who actively participated in the development of the standard. The same arguments that confronted the IEEE standard developers probably will come up again when a seminar participant tries to incorporate the standard into his or her organization. An instructor who can share experiences with these problems and their solutions is invaluable.

It is highly desirable for instructors to be experienced in performing the jobs associated with the standard. A standard may say, for example, "Define the approach used in testing," or "Define the functional requirements of the system."

An instructor who can share experiences with these problems and their solutions is invaluable.

A student's first reaction is to ask, "What are the available approaches to testing?" or, "How do I go about defining requirements?"

To answer these questions from experience on "how it worked on project X", requires an instructor who has used that knowledge on project X. Knowledge derived from reading a textbook will not satisfy the average IEEE software standards seminar participant. In fact, most people who come to the seminars want to know not only where the technology was used but what was wrong with it. The only instructor who can survive such bombardments of questions is one with experience.

All of the standards development knowledge and job experience in the world will not compensate for poor communications skills. So, instructors for the IEEE software standards seminars are people who have histories of public speaking successes.

Materials. In our discussions of limiting subject matter, we touched on how the
seminar materials are organized and kept up-to-date. But how are the materials developed?

An IEEE software standards seminar is developed in phases. Each phase terminates with a deliverable product that is reviewed for completeness, correctness, and consistency before the next phase is initiated. At every step, a consensus must be reached by all members of the developmental/instructional team and the seminar chairperson for the IEEE Computer Society Standards Activities Board that the product of this phase is acceptable. Occasionally, instructors from other seminars or chairpersons of developing standards also will be asked to participate in the review.

The old, familiar consensus approach coupled with the latest techniques from the field of instructional technology helped evolve five phases called defining, designing, implementing, testing, and maintaining. These phase names should be recognizable by anyone developing software today. These partitions of the life cycle work well for seminar development, too.

In the defining phase, the team will produce a seminar requirements document. This includes a description of the need for the seminar (needs assessment), the expected audience, the knowledge the student is expected to walk away with (educational objective), and the allowable sequences for presentation of the knowledge (educational objectives hierarchy). In the designing phase, the team will create a high-level description of how information will flow throughout the entire seminar (architectural design). The team will also produce a description of information flow in each instructional module (detailed design). Also, in this phase attention is paid to developing hands-on exercise modules.

Finally, in the implementing phase, the instructors get to produce some viewgraphs (do some coding). In this phase these visual aids will be hand-drawn first cuts.

The team usually moves into the testing phase with a sense of excitement and anticipation. This is when formal trial runs of the entire seminar are conducted. The audience usually will include subject matter experts, instructional technologists, and typical students who were not involved in the development of the seminar.

The maintaining phase is a final and ongoing process. In this phase, the seminar is offered through open enrollment delivery. Auditors from the IEEE standards office drop in on presentations. Seminar participants are asked to provide formal feedback to the instructors after each seminar, and all this information is considered in the next maintenance release.

We can see that guidelines and controls are important in establishing first-rate professional seminars. But, most of the success of these seminars must be credited to the dedicated efforts of the many IEEE members involved with the standards process and to the companies across the country who are supporting these efforts. A special thanks is owed to the seminar instructors who give a great deal of time and effort to help develop the standards and then give still more to develop and teach the seminars. The names of our current instructors, along with their company and seminar affiliations, are listed in the accompanying box. A fall seminar schedule is listed there as well.

Please direct any technical questions or comments about the seminars to Bob Poston. Inquiries about registration should be made to Susan Havranek at IEEE headquarters in New York. The phone number is (212) 705-7907.

---

Instructors and schedules
Instructors for the current IEEE software standards seminars are:

- A. Frank Ackerman
  Software Engineering Associates
  Berkeley Heights, New Jersey
  (software requirements specifications)

- Eleanor Antreasian
  Computer Sciences Corporation
  Piscataway, New Jersey
  (software testing)

- H. Ronald Berlack
  Sanders Associates, Inc.
  Nashua, New Hampshire
  (software configuration management)

- Fletcher J. Buckley
  RCA
  Moorestown, New Jersey
  (software quality assurance)

- Alan M. Davis
  BTG, Inc.
  Vienna, Virginia
  (software requirements specifications)

- David Gelpin
  Software Quality Engineering
  Minneapolis, Minnesota
  (software testing)

- Thomas L. Hannan
  Computer Technology Associates
  Denver, Colorado
  (software quality assurance)

- Dwayne L. Knirk
  Programming Environments, Inc.
  Oakhurst, New Jersey
  (software configuration management)

- George D. Tice
  Tektronix, Inc.
  Wilsonville, Oregon
  (software configuration management)

Robert M. Poston is seminar chairperson for the IEEE Computer Society Standards Activities Board. He may teach various subject seminars during a season to assure consistency across all seminars.

A schedule of the IEEE software standards seminars is listed below:

<table>
<thead>
<tr>
<th>Seminar</th>
<th>Date</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software quality assurance</td>
<td>Sep. 11-13</td>
<td>San Francisco, California</td>
</tr>
<tr>
<td></td>
<td>Sep. 30, Oct. 1, 2</td>
<td>Chicago, Illinois</td>
</tr>
<tr>
<td></td>
<td>Oct. 21-23</td>
<td>Washington, D. C.</td>
</tr>
<tr>
<td>Software testing</td>
<td>Sep. 9, 10</td>
<td>San Francisco, California</td>
</tr>
<tr>
<td></td>
<td>Oct. 22, 23</td>
<td>Washington, D. C.</td>
</tr>
<tr>
<td></td>
<td>Nov. 5, 6</td>
<td>Orlando, Florida</td>
</tr>
<tr>
<td>Software configuration management</td>
<td>Oct. 24, 25</td>
<td>Washington, D. C.</td>
</tr>
<tr>
<td></td>
<td>Nov. 7, 8</td>
<td>Orlando, Florida</td>
</tr>
<tr>
<td></td>
<td>Dec. 2, 3</td>
<td>San Diego, California</td>
</tr>
<tr>
<td>Software requirements specifications</td>
<td>Oct. 3, 4</td>
<td>Chicago, Illinois</td>
</tr>
<tr>
<td></td>
<td>Oct. 24, 25</td>
<td>Washington, D. C.</td>
</tr>
<tr>
<td></td>
<td>Dec. 4, 5</td>
<td>San Diego, California</td>
</tr>
</tbody>
</table>

---