THE IEEE SOFTWARE ENGINEERING STANDARDS PROCESS

Fletcher J. Buckley
RCA Government Systems Division
Moorestown, New Jersey

Abstract

Software Engineering has emerged as a field in recent years, and those involved increasingly recognize the need for standards. As a result, members of the Institute of Electrical and Electronics Engineers (IEEE) formed a subcommittee to develop these standards. This paper discusses the ongoing standards development, and associated efforts.

CR Categories and Subject Descriptors: D.2.0 [Software Engineering]: General -- standards

Introduction

The purpose of this paper is to relate the development of the norms of professional practice in the field of Software Engineering and to describe its interaction with the volunteer standards-making process of the Institute of Electrical and Electronics Engineers (IEEE). Towards this end, this article is presented in four sections:

1. Section 1 provides an overview of the environment in which the IEEE is currently producing Software Engineering Standards.
2. Section 2 describes the current thrust of the Software Engineering Standards effort.
3. Section 3 provides an overview of the seminars used to explain these standards and promote their use.
4. Section 4 relates the efforts currently used to avoid inbreeding in the standards-formulation efforts.

Section 1. The IEEE Software Engineering Standards-Making Process

This section provides an overview of the policies, operating procedures, and methods that have evolved for the use of the Software Engineering Subcommittee over the past six years. In addition, this section attempts to indicate the rationale for these policies and the lessons learned during the course of these efforts to date.

Standards are developed by working groups, balloted by the Software Engineering Subcommittee, and approved by the IEEE Standards Board. Recognizing the opportunities to go astray administratively, the orientation is to avoid administrative pitfalls that could delay the approval process.

Figure 1 identifies the environment in which the Software Engineering Standards Subcommittee (SESS) is embedded. As shown in this figure:[1]

a. The American National Standards Institute (ANSI) acts as the umbrella organization for all volunteer standards-making organizations in the United States. Details of ANSI are provided in the ANSI Constitution and Procedures [1] [2].

b. The Institute of Electrical and Electronics Engineers (IEEE) operates in part under the umbrella of ANSI as one of several volunteers standards-making organizations. As an international organization, the IEEE also operates a portion of its standards-making functions outside of ANSI. The major activity of the IEEE with which this guide is concerned is the IEEE Standards Board. Details of the IEEE organization are provided in the IEEE Constitution and Bylaws while the detailed requirements of the IEEE Standards Board are provided in the IEEE Standards Manual [3] [4] [5].

c. The IEEE Computer Society is one of the member societies of the IEEE. Details of the IEEE Computer Society organization are provided in the IEEE Computer Society Constitution and Bylaws [6] [7].

d. As established in Article XI of the IEEE Computer Society Bylaws, the Standards Activities Board monitors all Standards Activities to assure conformance to approved policies and practices. One specific activity of the Standards Coordinating Committee of direct interest to all is the Computer Dictionary Project (P610) [8].[2]

[1] For a more detailed examination of this figure and the associated processes, see Buckley, F. "A Guide to Standards Development for the Software Engineering Standards Subcommittee" *Computers and Standards*, 2 (1983) pp 185-199.

[2] For details on this project, contact should be made with the working group chairperson:

Jane Radatz
(619) 455-1330
Logicon, Inc
PO Box 80158
San Diego, Calif 92138

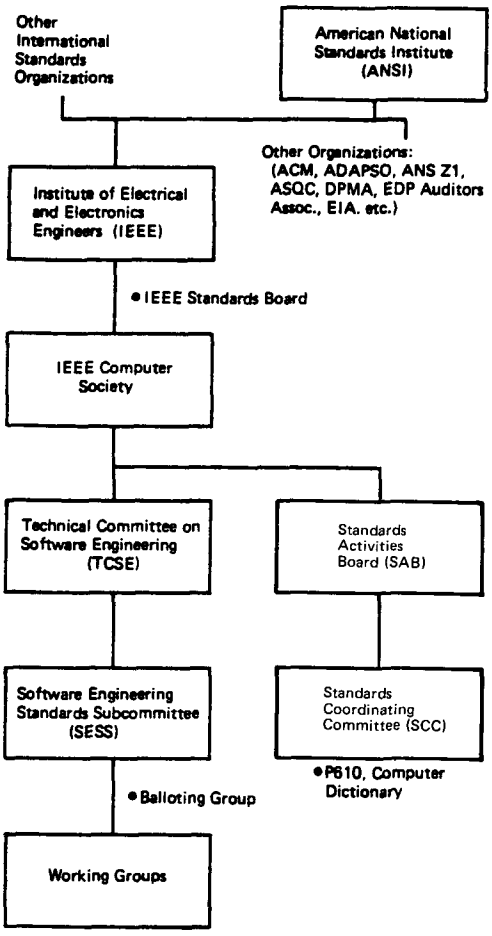

Figure 1. Standards-Making Organization

e. The Technical Committee on Software Engineering (TCSE) is one of several IEEE Computer Society Technical Committees that sponsor the development of standards. Details of the organization and activities of the TCSE are contained in the IEEE Computer Society Technical Activities Handbook [9].

f. The Software Engineering Standards Subcommittee (SESS) itself performs several activities:

   (1) It develops and ballots draft Software Engineering standards.

   (2) It co-sponsors Software Engineering Seminars with the IEEE Standards Board.

   (3) It organizes and runs the Software Engineering Standards Application Workshops.

The organizations that develop draft standards inside the SESS are called working groups, with one working group involved in developing one standard.

B. Working Group Membership

There are tradeoffs in the composition of the working groups. For instance, the goals of the project are to turn out the best product in an expeditious manner and have it approved by a broad consensus. These goals contain inherent conflicts.

One approach to turning out a product in an expeditious manner is to restrict the working group membership in some manner, for instance, to those who have participated by attendance or by providing technical input by letter, etc. The benefit of this approach is that the working group membership can be very efficient in producing a document. The difficulties with this approach are that:

(a) People who can not continually participate, do not provide input when they can; thus the document may not be as good as it could be

(b) A broad consensus is not developed as the document evolves; when ballotted by the Balloting Group, therefore, the document receives so many comments that it fails.

(c) Any arbitrary (unfair or nonobjective) restriction of membership is totally against IEEE policy, which seeks the development of a broad consensus. In addition, membership restriction could provide a basis for legal action for "restraint of trade." It is important not only to avoid arbitrary restrictions, but also to avoid actions that could provide a perception of such restrictions.

The approach that is believed better is to make the Working Group as large as possible and to actively encourage new members at all times. The benefits of this approach are that a substantial consensus develops as new inputs challenge old ideas. Also, there is substantial reinforcement of the Working Group's ideas as the old members bring the new ones up to date. The disadvantage to this approach is the increased time spent on the learning curve. One way to overcome this disadvantage is to start formally recording all comments at some time in the cycle, for instance, after the second complete draft. The comments are numbered and the disposition of each recorded together with the reasons why. These comments are attached to the minutes of the meeting. Based on the above, the decision has been made not to restrict working group membership.

C. Balloting Group Membership

To be a member of a Balloting Group requires one of three conditions to be met. Either 1) the person must be a member of the IEEE, 2) the person must be an Affiliate Member of the IEEE Computer Society, or 3) the Chairperson of the SESS can write a special letter to the IEEE Standards Board stating that the person being considered has such a degree of expertise that his or her participation is required. After that request is granted, the person then becomes a member of the Balloting Group.

Representatives from standards-making organizations may also ballot on draft standards. This requires a letter from that organization to the IEEE Standards Board (with copy to the Chairperson of SESS) requesting permission to participate and designating that person as the representative.

D. Processing of Draft Standards

When the working group has completed its efforts, the SESS Chairperson will run the ballot on the draft standard. The SESS Chairperson, in coordination with the Chairperson of the Working Group 1) resolves the comments from the balloting, 2) provides notice to the Balloting Group, and then 3) forwards the draft Standard to the IEEE Standards Board for approval. After approval, the IEEE Standards Board will normally forward the Standard to ANSI for adoption as an ANSI Standard.

Section 2. Extension of the Software Engineering Standards Effort

The overall goals for the SESS have been to help establish the norms of professional practice in the Software Engineering Field by means of the consensus process. Two trends have since been established for this work:

- Tree down from established standards for further development
- Respond to identification of needs by individuals.

A. Approved Standards

Table I identifies the Software Engineering Standards approved by the IEEE Standards Board to date.[3,4]

TABLE I. APPROVED SOFTWARE ENGINEERING STANDARDS

Standard	Title
ANSI/IEEE Std 729-1983	IEEE Standard Glossary of Software Engineering Terminology
ANSI/IEEE Std 730-1984	IEEE Standard for Software Quality Assurance Plans
IEEE Std 828-1983	IEEE Standard for Software Configuration Management Plans
ANSI/IEEE Std 829-1983	IEEE Standard for Software Test Documentation
IEEE Std 830-1984	IEEE Guide for Software Requirements Specifications

The first standard to be approved, ANSI/IEEE Std 730-1981, (since revised to 730-1984) has as its rationale, legal liability. It is directed towards the development and maintenance of critical software, that is, where failure could impact safety or cause large financial or social losses.

The standard establishes a required format and a set of minimum required contents for Software Quality Assurance Plans.

The second IEEE Software Engineering Standard to be approved, ANSI/IEEE Std 729-1983, establishes definitions for most of the Software Engineering terms in general usage. It contains definitions for more than five hundred terms and therefore establishes the basic vocabulary of software engineering.

[3]Copies of approved IEEE Standards are available from The IEEE Publications Center, 345 Hoes Lane, Piscataway, NJ 08854 USA

[4]This should not be construed as an interpretation of meaning of any of the specific contents of any of these standards. Official interpretations of IEEE standards can only be furnished by the IEEE Standards Board in writing in response to a formal request.

The third Software Engineering Standard to be approved, ANSI/IEEE Std 829-1983, defines the content and format for eight documents that cover the entire testing process. The purpose of the standard is to describe a set of basic software test documents which cover test planning, test specification, and test reporting.

The test plan prescribes the scope, approach, resources, and schedule of the testing activities. It identifies the items to be tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the risks associated with the plan.

Test specification is covered by three document types:

1. A test design specification refines the test approach and identifies the features to be tested by this design and the associated tests. It also identifies the test cases and test procedures, if any, required to accomplish the testing and specifies the feature pass/fail critieria.
2. A test case specification documents the actual values used for input along with the anticipated outputs.
3. A test procedure specification identifies all steps required to operate the system and provide the specified test cases to implement the associated test design.

Test reporting is covered by four document types:

1. A test item transmittal report identifies the test items being transmitted for testing in the event that separate development and test groups are involved or in the event that a formal beginning of test execution is desired.
2. A test log is used by the test team to record what occurred during test execution.
3. A test incident report describes any event that occurs during the test execution which requires further investigation.
4. A test summary report summarizes that testing activities with one or more test design specifications.

The standard shows the relationships of these documents to one another as they are developed and to the test process they document. The standard also contains four appendices.

Appendix A of the standard contains examples which are meant to clarify the intent of the document descriptions found in the standard. Some suggestions about implementing and using the standard appear in Appendix B. Appendix C contains references to related test documentation standards. Appendix D contains references to testing-related documents of general interest which are not focused on test documentation.

The fourth Software Engineering Standard to be approved is IEEE Std 828-1983. This Standard is similar in format to the Standard for Quality Assurance Plans but it deals with the more limited subject of Software Configuration Management. The Standard gives requirements for configuration identification, configuration control, configuration status accounting and reporting, and configuration audits and reviews. This provides a means for ensuring the integrity of the software product item as it evolves through the Software Development Cycle.

The fifth Software Engineering Standard to be approved is IEEE Std 830-1984, IEEE Guide for Software Requirements Specifications. This is issued as a guide because the current consensus on the state-of-the-art is that there is no one recommended way to write a software requirements specification at this time.

B. Current Efforts

Table II provides a list of the current approved projects. Table III identifies new initiatives in this field. Some of these efforts are just starting, others are nearing completion. Some provide guidance for implementing standards that have been completed, such as Guides for Software Quality Assurance and Software Configuration Management; others are breaking into new areas such as Design Descriptions and Software Unit Testing.[5]

C. Themes

Throughout all of this effort there are several themes, the first of which has been consensus. As reflected in the balloting statistics shown in Table IV, the SESS has been averaging over 90% of Ballot Returns and over 90% of the returned ballots have been Approvals.

This theme has been further carried forward into the evolution of the Software Engineering Standards efforts. Thus, the IEEE Standard for Software Configuration Management Plans evolves and expands from the IEEE Standard for Software Quality Assurance Plans, as does the Guide for Software Requirements Specifications, the draft Guide for Sotware Design Descriptions, the draft Standard for Software Verification Plans, and the draft Standard for Software Reviews and Audits.

---

[5]For further information on the current projects, contact should be made as follows:

a) 982. A Standard For Software Reliability Measurement:

Jim Dobbins
(703) 367-3912
MS 105-913
IBM FSD
9500 Goodwin Drive
Manassas, Va. 22110

b) 983. A Guide For Software Quality Assurance:

G. Tice
(503) 629-1310
Tektronix
PO Box 392
Wilsonville, Oregon 97070

c) 990. A Guide For The Use Of Ada* As a Program Design Language:

Bob Blasewitz
(609) 778-3955
RCA, MS 101-210
Moorestown, NJ 08057

d) 1002. Software Engineering Standards Taxonomy

Leonard Tripp
(206) 575-5390
Boeing Computer Services
MS 9C-70, PO Box 24346
Seattle, Washington, 98124

e) 1008. A Standard For Software Unit Testing

David Gelperin
(612) 541-1431
2425 Zealand Ave., N.
Golden Valley, Minn 55427

f) 1012. A Standard For Software Verification Plans

Roger Fujii
(213) 831-0611
Logicon, Inc.
255 West Fifth St.
San Pedro, Calif 90731

g) 1016. A Guide For Software Design Descriptions

H. Jack Barnard
(303) 538-3976
Mail Stop 1D30
Atmot Info. Sys. Labs.
11900 North Pecos St.
Denver, Colorado, 80234

h) 1028, A Standard For Software Reviews and Audits

Charles P. Hollocker
(312) 979-4137
Western Electric
901 Rolling Drive
Lisle, Ill. 60532

i) 1042, A Guide For Software Configuration Management

Richard Van Tillburg
(714) 732-2307
Hughes Aircraft Corporation
Bldg 618, MS B209
PO Box 3310
Fullerton, Calif 92634

j) 1044, A Standard Classification Of Software Errors, Faults, And Failures.

Dick Evans
(213) 536-3805
TRW, Mail Stop R4/2182
One Space Park
Redondo Beach, California 90728

k) 1045, A Standard For Software Productivity Metrics

Eleanor Antreassian
(201) 981-6479
Bell Labs
Room 3B121
6 Corporate Place
Piscataway, New Jersey 08854

l) A Guide For Third Party Software Acquisition

Philip C. Marriott
(513) 445-2198
NCR Corporation
World Hqs 4
Dayton, Ohio 45479

m) A Standard For Software Quality Metrics

Dr. Norman F. Schneidewind
(408) 646-2719/3211
Professor, Dept RSA/CS
Code 54SS
Naval Postgraduate School
Monterey, Calif 93940

n) A Standard For User Documentation

Christopher Cooke
(301) 338-5644
Martin Marietta Aerospace
Mail Station 98
103 Cheasapeake Park Plaza
Baltimore, Md 21220

TABLE II. APPROVED SOFTWARE ENGINEERING STANDARDS PROJECTS

Project	Title	Approved
P982	A Standard for Software Reliability Measurement	Dec. 1982
P983	A Guide for Software Quality Assurance	Dec. 1982
P990	A Guide for the Use of Ada* as a Program Design Language	Mar 1983
P1002	Software Engineering Taxonomy	June 1983
P1008	A Standard for Software Unit Testing	June 1983
P1012	A Standard for Software Verification Plans	Sept. 1983
P1016	A Guide for Software Design Descriptions	Sept. 1983
P1028	A Standard for Software Reviews and Audits	Mar. 1984
P1042	A Guide for Software Configuration Management	June 1984
P1044	A Standard Classification for Software Errors, Faults, and Failures	June 1984
P1045	A Standard for Software Productivity Metrics	June 1984

*Ada is a registered trademark of the US Government, AJPO

TABLE III. SOFTWARE ENGINEERING STANDARDS NEW INITIATIVES

Title
A Standard for Software Quality Metrics
A Standard for User Documentation
A Guide for Third Party Software Acquisition

TABLE IV. SOFTWARE ENGINEERING STANDARDS BALLOTING STATISTICS

Standard	Members of Balloting Group	Returns	Approve	Disapprove	Abstain
730-1981 (Trial-Use)	104	82(78.8%)	64(78.0%)	8(9.8%)	10(12.2%)
730-1981 (Full-Use)	120	102(85.0%)	86(84.3%)	—	16(15.7%)
730-1984 (Revision)	181	154(85.1%)	144(93.5%)	1(0.6%)	9(5.9%)
729-1983	147	138(93.9%)	128(92.8%)	1(.7%)	9(6.5%)
828-1983	114	105(92.1%)	99(94.3%)	1(1.0%)	5(4.7%)
829-1983	102	96(94.1%)	83(86.5%)	—	13(13.5%)
830-1984	102	90(88.2%)	82(91.1%)	1(1.1%)	7(7.8%)

The second evolution is from Standard to Guide. The view expressed here is that a Standard, defined in the restricted sense, is the first document to be produced, assuming that a consensus exists that such a document can be produced. A Standard should be followed by a Guide or a Recommended Practice. The Guide or Recommended Practice would contain tutorial material, examples, and sage advice — in other words, material not considered appropriate for a Standard. As such, the IEEE Standard for Software Quality Assurance Plans is being followed by a Guide, as is the Standard for Configuration Management Plans.

The last is the evolution from product standard to process standard. It is in this context, for example, that the process standard, the draft Standard on Software Unit Testing, has been initiated from the IEEE Standard on Software Test Documentation.

The second theme has been timeliness. The best standards in the world will not help if they are not provided in a timely manner. As depicted in Table V, the SESS is approaching a figure of three years from project approval to approval of the resulting standard.

TABLE V. SOFTWARE ENGINEERING STANDARDS PRODUCTION STATISTICS

Standard	Project Approved	Standard Approved	Elapsed Time
730-1981	Dec 1977	Dec 1979 (Trial-Use)	—
		Sept 1981 (Final)	3-3/4 years
730-1984	Sept 1982	June 1984 (Revision)	1-3/4
729-1983	March 1978	Sept 1982	4-1/2 years
828-1983	March 1980	June 1983	3-1/4 years
829-1983	March 1980	Dec 1982	2-3/4 years
830-1984	March 1980	Sept 1983	3-1/2 years

The third theme has been to recognize the need for underlying structure documents. Initially the IEEE Standard for Software Quality Assurance Plans and the IEEE Standard Glossary of Software Engineering Terminology served that need. To extend a complete structure, the Software Engineering Taxonomy effort has been initiated.

Concurrently with the third theme has been the emphasis of being responsive to the perceived needs of the Software Engineering community. This has led to initiation of specific efforts even though a full set of supporting standards and guides are not in place. These efforts include the draft Guide for Software Reliability Measurement, the draft Guide for Ada as a PDL, and the draft Standard Classification of Software Errors, Faults, and Failures. These are urgently needed and cannot wait for another three or four years until a full infrastructure is available.

Section 3. Software Engineering Standards Seminars

An important part of the Software Engineering Standards process is the acceptance of these standards by the communities of associated professionals. The best standards in the world are of little value if they are not adopted and used by the professional community for which they are aimed.

Since October 1979, the IEEE Standards board has been sponsoring Software Engineering seminars. These seminars are based on the approved standards and serve to provide familiarization and extensions to the particular documents involved.

Table VI provides a summary of the current seminars. It is planned to initiate at least two more in Fall of 1985.[6]

TABLE VI. SOFTWARE ENGINEERING STANDARDS SEMINARS

Seminar	Initiated	Notes
Software Quality Assurance	Fall '79	Currently in 4th Revision. Based on ANSI/IEEE Std 730-1984
Software Testing	Fall '83	Based on ANSI/IEEE Std 829-1983
Software Configuration Management	Fall '84	Based on IEEE Std 828-1983
Software Requirements Specifications	(Spring '85)	Based on IEEE Std 830-1984

Section 4. Software Engineering Standards Application Workshops

In any successful effort there is an almost overwhelming temptation to succumb to inertia, to inbreeding of ideas, to complacency. As one step to avoid these traps, a series of Software Engineering Standards Application Workshops (SESAW) have been initiated. Data on these workshops is provided in Table VII.

The critical review of the efforts of the Software Engineering Standards field from the last workshop led directly to the new initiatives in metrics and classifications.[7]

TABLE VII. SOFTWARE ENGINEERING STANDARDS APPLICATION WORKSHOPS STATISTICS

Title	Date	Papers	Attendees	Notes
SESAW-I	Aug 1981	22	72	Overall Emphasis on Stds Creation
SESAW-II	May 1983	35	110	Major Source of Guidance for future Software Engineering Standards Activities
SESAW-III	(October 1984)			

Looking Forward

The main motivation behind the creation of these IEEE standards has been to provide recommendations reflecting the current state of the art in the application of engineering principles to Software Engineering. These principles will continue to evolve, and these standards are meant to serve as starting points for further development. For those that are new to Software Engineering, the standards can be an invaluable source of carefully considered advice. For those that are on the leading edge of the field, the standards serve as a baseline against which advances can be communicated and evaluated.

These standards will ultimately define the norm of professional practice in all aspects of software development and maintenance from requirements definition through acceptance testing and beyond. They will be widely employed because they have been arrived at in a open process of professional discussion and debate. At any point in time, they represent the professional consensus on what should be done to produce the type of software our society now depends on.

References

1. *Constitution and Bylaws of the American National Standards Institute,* May 1978. (Copy available from American National Standards Institute (ANSI), 1430 Broadway, New York, NY, 10018).
2. *American National Standards Institute Procedures for Management and Coordination of American National Standards,* effective 1 September 1982.
3. *IEEE Constitution,* as amended to Dec 23, 1980. (Copy available from the IEEE, 345 East 47th St., New York, NY 10017).
4. *IEEE Bylaws,* January 1982
5. *IEEE Standards Manual,* Sept 1982. IEEE Standards Board, 345 East 47th St., New York (Sept. 1982).
6. *IEEE Computer Society Constitution* (Copy available from IEEE Computer Society Headquarters, PO Box 639, Silver Springs, Md. 20901)
7. *IEEE Computer Society Bylaws*
8. *IEEE Computer Society Standards Development Guide,* March 3, 1983.
9. IEEE Computer Society, *Technical Activities Board Handbook.*

[6]For details on these seminars contact should be made with S. Havranek, Marketing Manager, IEEE Standards Board, 345 East 47th St., N.Y, N.Y. 10017. (Telephone (212) 705-7907).

[7]For a comprehensive review of SESAW-II, see Gloss-Soler, S "Report of SESAW-II" TCSE Newsletter, Winter 1983. (Copy available from IEEE Computer Society Headquarters, PO Box 639, Silver Spring, Md 20901.)