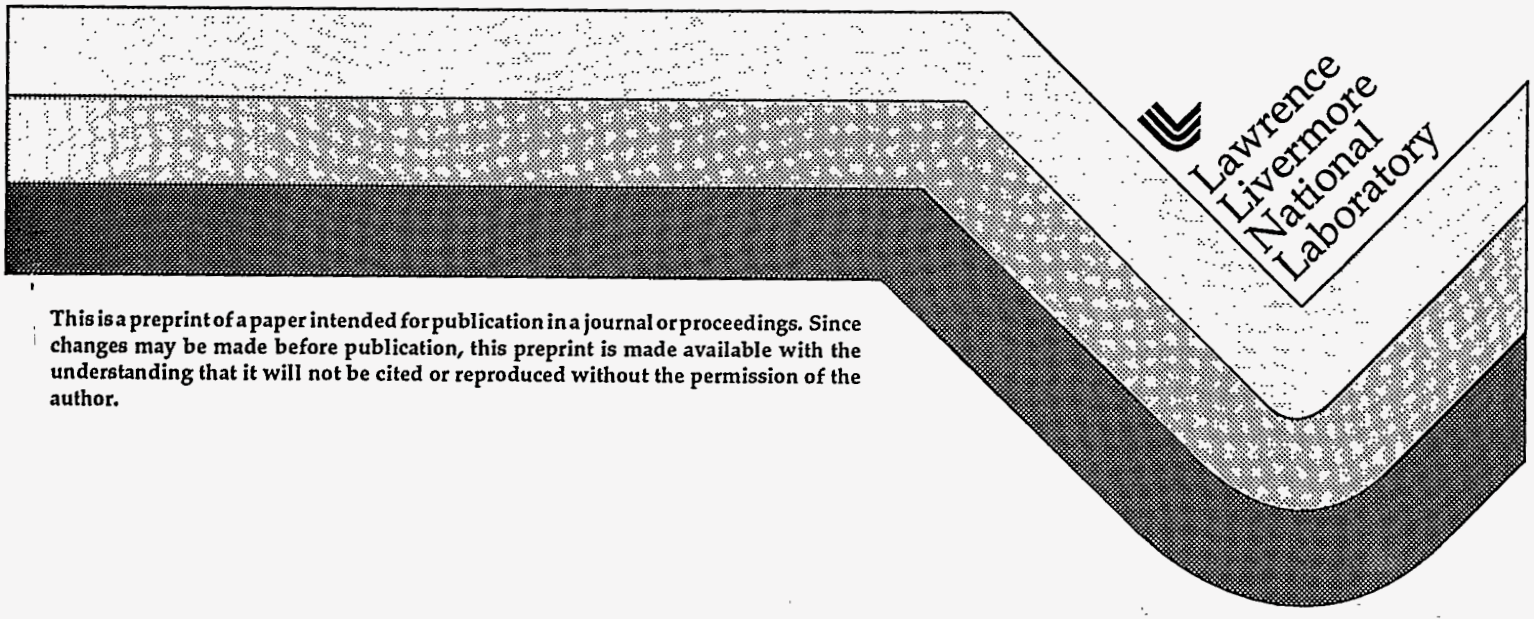
UCRL-JC-122249
PREPRINT
CONF-9511160--2

An Overview of Software Safety Standards

J. Dennis Lawrence

This paper was prepared for submittal to the Second
International Federation of Automatic Control (IFAC) Workshop on
Safety and Reliability in Emerging Control Technologies
Daytona Beach, Florida
November 1-3, 1995

October 1995



Disclaimer

This work was supported by the United States Nuclear Regulatory commission under a Memorandum of Understanding with the United States Department of Energy, and performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

AN OVERVIEW OF SOFTWARE SAFETY STANDARDS*

J. Dennis Lawrence

University of California
Computer Safety & Reliability Group
Fission Energy and Systems Safety Program
Lawrence Livermore National Laboratory
7000 East Avenue, L-632
Livermore, CA 94550
e-mail address: lawrence2@llnl.gov

Abstract: The writing of standards for software safety is an increasingly important activity. This essay briefly describes the two primary standards-writing organizations, IEEE and IEC, and provides a discussion of some of the more interesting software safety standards.

Keywords: Computer software, safety, standards.

1. INTRODUCTION

The use of computers in safety-critical applications has increased considerably over the last several years, and is expected to continue at an increasing rate into the near future. Application areas include power plants, medical devices, aircraft, chemical plants, automobiles, and military weapons. Accompanying this increased usage is an increasing number of standards that can be used to help create safety-critical software systems.

Approximately three or four new standards have been appearing each year for most of the past decade, with several dozen now available. These are produced by a large variety of organizations, including professional societies, governments and individual companies.

This essay provides a brief survey of standards that relate directly to software safety. No claim is made for completeness; the author is not aware of all organizations writing such standards. Standards change as old versions are updated and new documents are written.

2. STANDARDS-WRITING ORGANIZATIONS

There are at least three dozen organizations writing standards that relate to software. Over a dozen of these have written software standards that directly relate to safety. Both of these numbers are likely to be low.

Standards organizations are affiliated with a variety of groups, from broad professional societies to individual companies. Table 1 provides a partial list of such organizations, divided into multi-disciplinary societies, governments, and industry sector societies. Individual companies that have written software safety standards are not included in this list.

The most important organizations are probably the Institute of Electrical and Electronics Engineers (IEEE) and the International Electrotechnical Commission (IEC). These are discussed next.

2.1. IEEE

IEEE is a large organization composed of societies that cover most areas of electrical and electronics engineering. One such society is the Computer Society, a portion of whose structure is outlined in Figure 1.

*UCRL-JC-122249. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract no. W-7405-Eng-48.

The Computer Society is divided into technical activities, governed by the Technical Activities Board; standards activities, governed by the Standards Board; and other activities. One technical committee (TC) covers Software Engineering and one standards committee (SC) is the Software Engineering Standards Committee (SESC). These two committees work together to write standards on software engineering. As shown in Figure 1, standards activities report both to the Computer Society and to the IEEE Standards Board that oversees all IEEE standards work.

The Computer Society has written several dozen software engineering standards. One of these, IEEE 1228, Standard for Software Safety Plans, relates to software safety. In 1991, SESC began developing a long-range plan for software engineering standards. This plan (IEEE 1993), approved in December 1993, calls for additional work on software safety standards. A planning group, the Software Safety Planning Group (SSPG), was created in May 1995 to write a detailed plan for this topic.

Software engineering standards written by IEEE are generally submitted to standards-approving bodies for endorsement. In the United States, this is the American National Standards Institute (ANSI).

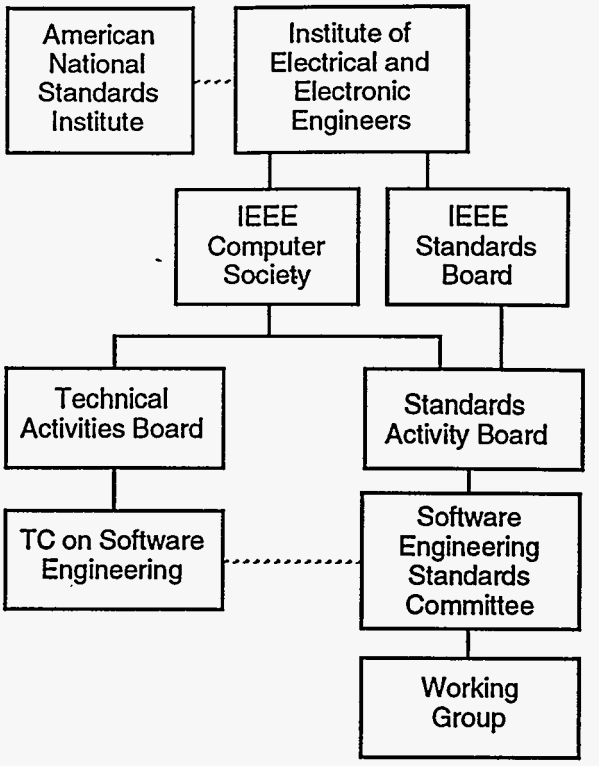


Figure 1. Structure for IEEE Software Engineering Standards Efforts

Table 1. Some Organizations Writing Software Safety Standards

Multi-disciplinary Societies	
BCS	British Computer Society
IEC	International Electrotechnical Commission
IEE	Institute of Electrical Engineers
IEEE	Institute of Electrical and Electronics Engineers
Government Bodies	
CSA	Canadian Standards Association
DIN	Deutsches Institut fur Nomrung e.V.
DOD	United States Department of Defense
MOD	United Kingdom Ministry of Defence
NASA	National Aeronautics and Space Administration
Industry Sector Societies	
ANS	American Nuclear Society
ASME	American Society of Mechanical Engineers
EIA	Electronics Industry Association
ISA	Instrument Society of America
RTCA	Requirements and Technical Concepts for Aviation
UL	Underwriter's Laboratories

2.2 IEC

IEC is an organization composed of countries interested in standards activities in the areas of electricity, electronics and associated technologies. The technical work of writing standards is governed by the Committee of Action (Figure 2). Several of the 87 technical committees are involved in software standards, safety standards, and software safety standards.

TC 65 is responsible for industrial-process measurement and control; it has a subcommittee, SC 65A, which is responsible for systems aspects. This subcommittee has been working on safety standard IEC 1508 for many years.

IEC also contains industry-specific technical committees, some writing standards related to software safety. An example is TC 45, Nuclear instrumentation, and SC 45A, Reactor instrumentation, responsible for several standards relating to the safety of nuclear reactors when software is involved in reactor safety.

IEC works closely with the International Organization for Standards (ISO). The two organizations sponsor Joint Technical Committee 1, which works on matters of information technology, including software engineering.

3. EXAMPLES OF SOFTWARE SAFETY STANDARDS

Table 2 lists a number of existing and proposed standards directly related to software safety. Since any aspect of software engineering may potentially affect safety, a great number of other standards exist that are indirectly related to safety. They are generally considered beyond the scope of this essay.

The remainder of this section consists of comments on a few standards listed in Table 2. Comments are based on the author's personal interests and opinions.

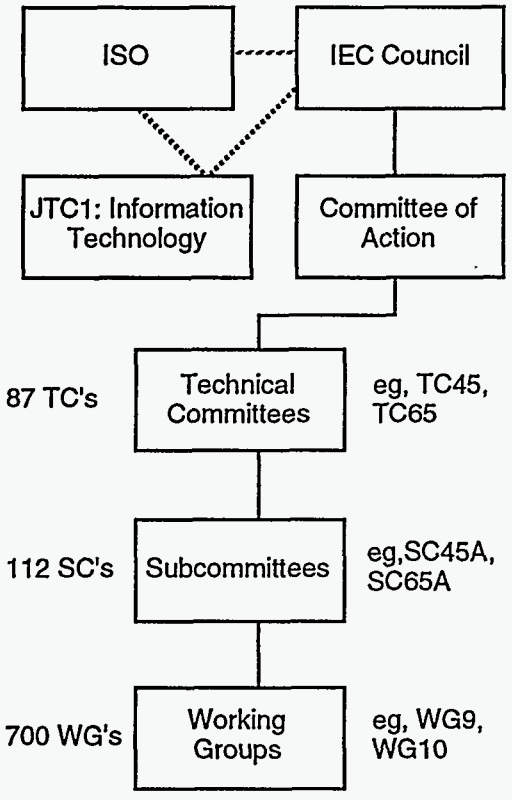


Figure 2. Structure for IEC Software Safety Standards Efforts

3.1. AIAA R-013, Recommended Practice for Software Reliability

Software reliability is not identical to safety, but is certainly a prerequisite. This standard can be used to create and operate a program for software reliability estimation. The purpose is to obtain reasonably precise, quantitative estimates of the reliability of software products.

3.2. ASME NQA-1, Quality Assurance Requirements for Nuclear Facility Applications

This standard has undergone many revisions since it was first issued in 1979. The latest version incorporates both NQA-1 and NQA-2, and covers most aspects of nuclear facility quality assurance. Of particular interest is Part II, subpart 2.7, "Quality Assurance Requirements for Computer Software for Nuclear Facility Applications."

NQA-1 contains a wealth of ideas that relate to quality assurance, and thus to facility safety. Much of this can be usefully transferred to other process-control applications where safety is involved.

3.3. Mil Std 882C, System Safety Program Requirements

Perhaps the most useful part of this standard is the concept of analyzing hazards by severity of potential consequences and probability of occurrence. These ideas can be combined into a table that can be used to assess risk; several examples are shown in Appendix A of the standard.

3.4. IEC 880, Software for Computers in the Safety Systems of Nuclear Power Stations

This standard describes principles and practices for the entire software development life cycle. Although directed at nuclear power stations, most of the concepts and specific details should apply equally to other process-control applications. Appendices include important recommendations on software development in safety-critical applications. A recent (1994) draft supplement discusses four additional topics: diversity against common mode failures, formal specification and design methods, automatic tools, and the use of pre-existing software.

3.5. Draft IEC 1508, Functional Safety: Safety-Related Systems

IEC SC 65A has been writing a standard on safety for some years. The result is a seven-part draft that covers much of system and software safety. The seven parts are: (1) General requirements, (2) requirements for electrical/electronic/programmable electronic systems, (3) software requirements, (4) definitions and abbreviations of terms, (5) guidelines on the application of part 1, (6) guidelines on the application of parts 2 and 3, and (7) bibliography of techniques and measures.

3.6. IEEE 1228, Standard for Software Safety Plans

This standard provides a suggested table of contents for a software safety plan, and discusses the content of each section, including software safety management, software safety analyses, and post-development procedures. Writing a safety plan, and then following the plan during software development and operation, should increase the probability of a successful and safe product.

Table 2. Examples of Software Safety Standards

Source	Number	Title	Date
AIAA	R-013	Recommended Practice for Software Reliability	1992
ASME	NQA-1	Quality Assurance Program Requirements for Nuclear Facilities	1994
ASTM	E 1246	Standard Practice for Reporting Reliability of Clinical Laboratory Computer Systems	1988
BCS	81205	System Safety Instruction—System Safety Engineering in Software Development	1989
CSA	Q396.1.1	Quality Assurance Program for the Development of Software Used in Critical Applications	1989
CSA	Q396.1.2	Quality Assurance Program for Previously Developed Software Used in Critical Applications	1989
DIN	V/VDE 0801	Principles for Computers in Safety-Related Systems (preliminary std)	1989
DOD	AFISC SSH 1-1	Software System Safety	1985
DOD	Mil-Hdbk 764	System Safety Engineering Design Guide for Army Materiel	1990
DOD	882C	System Safety Program Requirements	1993
EIA	SEB6-A	A Method for Software Safety Analysis	1990
ESA	PSS-01-40	System Safety Requirements for ESA Space Systems and Associated Equipment	1988
IEC	812	Analysis Techniques for System Reliability—Procedure for Failure Modes and Effects Analysis (FMEA)	1985
IEC	880	Software for Computers in the Safety Systems of Nuclear Power Stations	1986
IEC	987	Programmed Digital Computers Important to Safety for Nuclear Power Stations	1989
IEC	1014	Programmes for Reliability Growth	1989
IEC	1025	Fault Tree Analysis	1990
IEC	1226	The Classification of Instrumentation and Control Systems Important to Safety for Nuclear Power Plants	1993
IEC	1508	Functional Safety: Safety-Related Systems (draft)	1995
IEE	5	Software in Safety-Related Systems	1989
IEEE	7-4.3.2	Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations	1993
IEEE	1228	IEEE Standard for Software Safety Plans	1994
ISA	SP 84	Programmable Electronic Systems for Use in Safety Applications (draft)	1994
MOD	00-55	The Procurement of Safety Critical Software in Defence Equipment Part 1: Guidance	1991
MOD	00-56	The Procurement of Safety Critical Software in Defence Equipment Part 2: Requirements	1991
NASA	1740.13	Software Safety Standard	1994
RTCA	DO-178B	Software Considerations in Airborne Systems and Equipment Certification	1992
UL	1998	Standard for Safety-Related Software (draft)	1992

3.7. RTCA DO178-B, Software Considerations in Airborne Systems and Equipment Certification

This is the third version of a standard that provides guidance on determining whether software in aircraft meets the safety requirements of the aircraft. The increasing development of fly-by-wire aircraft increases the need to both develop software that (1) supports flight worthiness and (2) demonstrates that the

software supports flight worthiness. This standard can be of use for both goals.

4. CONCLUSION

Since an enormous number of organizations involved in software standards works around the world, the lack of coordination among them results in confusing and inconsistent results. In many ways, this has been a source of strength as multiple viewpoints emerge naturally and can be tested by groups attempting to use the standards. The author expects this to continue as more organizations become interested in writing standards for software in safety-critical applications.

Standards can be of considerable assistance to software developers, users and government regulatory bodies. Their use can provide a common vocabulary and common framework for expressing a software design and implementation, to the benefit of all parties. Standards are not a substitute for clear careful thinking, and no use of standards can guarantee safety. They are best thought of as one of many tools which can be used to help improve the safety properties of a software product.

REFERENCES

IEEE. 1993. *Master Plan for Software Engineering Standards*, Prepared by Software Engineering Standards Long-Range Planning Study Group, IEEE Software Engineering Standards Committee, December.

Technical Information Department • Lawrence Livermore National Laboratory
University of California • Livermore, California 94551