# A COMPARISON OF FIVE APPROACHES TO SOFTWARE DEVELOPMENT

David J. Schultz

January 21, 2000

## 1. Introduction

This white paper addresses five approaches, or methodologies, for software engineering (SWE):

- The IEEE Software Engineering Standards
- ISO/IEC/IEEE/EIA 12207
- The SEL Recommended Approach
- The SSDM Standards and Procedures
- ISO 9001 (with the accompanying guidance of ISO 9000-3).

(Because this is an informal white paper, many acronyms are not defined in the text. A glossary of acronyms, however, is included.)

The purpose of this white paper is to provide a tutorial about these five approaches to SWE. We shall identify some similarities and differences among these approaches. We shall also identify the target audience for each approach, describe the coverage of each approach, and indicate areas of overlap among the approaches.

This report comprises eight sections. Sections 2-6 summarize the five approaches, one by one. Section 7 contains an analysis of the five approaches, including similarities, differences, areas of overlap, and the potential applicability of each approach to current software work at the Information Systems Center (ISC). Section 8 contains a brief description of the associated research project.

## 2. IEEE Software Engineering Standards

### 2.1 History

The IEEE Software Engineering Standards effort began in 1976, with the creation of the Software Engineering Standards Subcommittee (SESS) of the Technical Committee on Software Engineering (TCSE). Three years later, the first IEEE SWE standard was published on a trial use basis. By 1984, there were five SWE standards:

- IEEE Std 729-1983, “IEEE Standard Glossary of Software Engineering Terminology”
- IEEE Std 730-1981, “IEEE Standard for Software Quality Assurance Plans”
- IEEE Std 828-1983, “IEEE Standard for Software Configuration Management Plans”
- IEEE Std 829-1983, “IEEE Standard for Software Test Documentation”
- IEEE Std 830-1984, “IEEE Recommended Practice for Software Requirements Specifications”

These original five standards have since been revised, some of them several times. IEEE 729 has since been redesignated IEEE Std 610.12-1990. In 1991, IEEE published a Software Engineering Standards Collection that included all of the currently approved SWE standards; there were about 16 standards by then. By 1994, the Standards Collection had grown to include 27 standards. The current edition of the Standards Collection, published in 1999, comprises four volumes and 40 standards. It also organizes the standards into an overall framework, and aligns the process and data standards with ISO/IEC 12207, the international standard on software life cycle processes. In the normal course of events, the American National Standards Institute (ANSI) eventually adopts each IEEE SWE standard. ANSI has adopted most of the standards in the 1999 Collection, although in some cases they have not yet adopted the most current version of a standard.

### 2.2 Scope

The 40 standards in the current collection “cover a broad scope of subjects related to the responsible and effective practice of the software engineering discipline.” The subject areas covered include terminology, software acquisition, life cycle processes, project management, plans, documentation, reuse, tools, and measurement.

### 2.3 Purpose

Although each individual standard was written to address a specific need, there has never been a single overall purpose for the entire collection. The collected standards do not fall into a neat hierarchy, and there are important topics that remain to be addressed. The challenge has been to organize the standards in a manner that facilitates the selection of the precise standard(s) appropriate for an individual user. For this reason, a framework has been developed for organizing and presenting the standards in the collection.

### 2.4 Organization

The SESC framework organizes the SWE standards into four categories, according to an object-oriented scheme. Software engineering tasks are generally considered to be performed by a project. This project interacts with its *customers*. It employs *resources* to perform *processes* and produce *products*. SESC has organized its collections of standards around these four objects; we thus have customer standards, resource standards, process standards, and product standards. Each volume of the Standards Collection contains the standards that relate to one of these four objects.

The standards are also organized according to how prescriptive they are, based upon the maturity of the corresponding discipline. IEEE develops three types of standards:

- Standards (with a capital “S”)
- Recommended Practices
- Guides.

Standards are prescriptive, contain requirements for conformance, and generally employ the verb ‘shall’. Recommended Practices present approaches that the IEEE recommends, but are used in disciplines for which no single approach is well enough accepted that IEEE can reach consensus on requiring it. They generally employ the verb ‘should’. Guides can suggest several alternative approaches to good engineering practice, but generally refrain from making specific recommendations; they traditionally use the verb ‘may’.

### 2.5 Audience

There is not a single audience for the entire collection. The 40 component standards have different audiences: project managers, system engineers, software developers, testers, CM personnel, QA personnel, system acquirers, etc. One of the reasons for partitioning the software collection into four volumes was to group together related standards that were likely to relate to the same audience.

### 2.6 Application Domain

Similarly, there is not a single application domain. Some of the subject areas addressed by the standards in Volume 1 of the collection (Customer and Terminology Standards), for example, are listed below:

- Software Acquisition
- Software Safety

- System Requirements
- Software Life Cycle Processes (IEEE/EIA 12207).

Some of the subject areas addressed by the standards in Volume 2 (Process Standards) are as follows:

- Software Quality Assurance
- Software Configuration Management
- Software Unit Testing
- Software Verification and Validation
- Software Maintenance
- Software Project Management
- Software Life Cycle Processes.

Some of the subject areas addressed by the standards in Volume 3 (Product Standards) are as follows:

- Measures to Produce Reliable Software
- Software Quality Metrics
- Software User Documentation.

Some of the subject areas address by the standards in Volume 4 (Resource and Technique Standards) are listed below:

- Software Test Documentation
- Software Requirements Specifications
- Software Design Descriptions
- Concept of Operations for Interoperating Reuse Libraries
- Evaluation and Selection of CASE Tools.

### 2.7 Applicability to State of the Practice

IEEE has a vested interest in maintaining the currency of the Standards Collection. IEEE requires that each individual standard be reviewed at least every five years to verify its continued applicability, and to determine whether the standard should be reaffirmed, revised, or withdrawn. Although IEEE has not met this five-year milestone for every standard, only two of the standards in the Collection date from earlier than 1993.

## 3. ISO/IEC/IEEE/EIA 12207

### 3.1 History

The International Organization for Standardization (ISO—this is by intention not an acronym) and the International Electrotechnical Commission (IEC) share responsibility for the development of international standards. Both ISO and IEC are comprised of national bodies that work through technical committees to develop standards for particular fields of activity. To address the field of information technology, ISO and IEC established a joint technical committee, ISO/IEC Joint Technical Committee 1 (JTC1). JTC1 spawned multiple subcommittees dedicated to preparing different types of information technology standards. The subcommittee chartered to prepare software engineering standards is designated Subcommittee 7 (ISO/IEC JTC1/SC7).

In the early 1990s, SC7 decided to write an international standard for software life cycle processes (SLCPs). The standard was published in 1995 as ISO/IEC 12207:1995, “Information technology—Software life cycle processes.” Among the national bodies that belong to SC7, there was a strong desire to use SLCPs in a contract-driven environment. ISO/IEC 12207 therefore has a strong emphasis upon the supplier/acquirer relationship. It specifies certain activities to be performed by the software development organization, and other activities to be performed by the acquiring organization.

In 1996, IEEE and the Electronic Industries Association (EIA) collaborated to publish a U. S. industry implementation of 12207. This document is designated as IEEE/EIA 12207 and is published in three volumes. IEEE/EIA 12207.0 is a reprinting of ISO/IEC 12207 with an IEEE/EIA cover page. IEEE/EIA 12207.1 provides content and format guidance, similar to military Data Item Descriptions (DIDs), for use in preparing project documents. IEEE/EIA 12207.2 provides guidance for implementing the specific process requirements of 12207.0.

### 3.2 Scope

ISO/IEC 12207 "applies to the acquisition of systems and software products and services, to the supply, development, operation, and maintenance of software products, and to the software portion of firmware, whether performed internally or externally to an organization. Those aspects of system definition needed to provide the context for software products and services are included.

"NOTE—the processes used during the software life cycle need to be compatible with the processes used during the system life cycle.

"This International Standard is intended for use in a two-party situation and may be equally applied where the two parties are from the same organization. The situation may range from an informal agreement up to a legally binding contract. This International Standard may be used by a single party as self-imposed tasks.

"This International Standard is not intended for off-the-shelf software products unless incorporated into a deliverable product."

### 3.3 Purpose

ISO/IEC 12207 defines a framework for software life cycle processes. It contains a hierarchy of processes, activities, and tasks to be applied in an environment where software is being developed, supplied, acquired, operated, and maintained. It addresses both the technical and the management aspects of software development and acquisition. Although originally written in English, the vocabulary employed has been carefully chosen so as to translate accurately into many other languages.

### 3.4 Organization

ISO/IEC 12207 covers three types of SLCPs: primary, supporting, and organizational. They are organized according to the following hierarchy:

- Primary Life Cycle Processes
  - Acquisition
  - Supply
  - Development
  - Operation
  - Maintenance

- Supporting Life Cycle Processes
  - Documentation
  - Configuration Management
  - Quality Assurance
  - Verification
  - Validation
  - Joint Review
  - Audit

- Organizational Life Cycle Processes
  - Management
  - Infrastructure
  - Improvement
  - Training

### 3.5 Audience

“This International Standard is written for acquirers of systems and software products and services and for suppliers, developers, operators, maintainers, managers, quality assurance managers, and users of software products.”

### 3.6 Application Domain

The application domain is defined by the component processes. See the hierarchy in 3.4, ‘Organization’.

### 3.7 Applicability to State of the Practice

ISO and IEC, like IEEE, have a vested interest in maintaining the currency of 12207. Both the international version and the IEEE/EIA version are to be reviewed every five years to ensure that they remain current. The next revision is therefore scheduled for 2001.

## 4. SEL Recommended Approach

### 4.1 History

As of this writing the SEL Recommended Approach has been published in three versions, as follows:

| | |
|---|---|
| SEL-81-105 | May 1982 |
| SEL-81-205 | April 1983 |
| SEL-81-305 | June 1992 |

According to the Introduction, the May 1982 version presents “recommendations for a disciplined approach to software development, with special emphasis on management considerations.” The first revision (April 1983) retains the same overall organization of the document, although the section on management was extensively revised. For the second revision, titled “Revision 3” (June 1992), a joint GSFC/CSC working group rewrote and reorganized the entire document. The management guidance in Section 4 of the 1983 revision was deleted from the 1992 revision and put into a separate Manager’s Handbook.

### 4.2 Scope

The document addresses software engineering projects in a flight dynamics environment. The document covers the following life-cycle phases: requirements definition, requirements analysis, preliminary design, detailed design, implementation, system testing, and acceptance testing. Maintenance and operation are beyond the scope of the 1992 revision; a few references to SEL papers on software maintenance, however, are included. (On page 10 of the document, there is an observation that the kinds of errors found during operations and maintenance are generally similar to those discovered during initial development, and that the associated corrections proceed through a development life cycle. On this basis, the text suggests that the Recommended Approach may be applicable to operations and maintenance after all.)

On page 3 of the Recommended Approach is the following caveat: “As these studies are confined to flight dynamics applications, readers of this document are cautioned that the guidance presented here may not always be appropriate for environments with significantly different characteristics.”

### 4.3 Purpose

According to the Introduction of the 1992 revision, “This document presents a set of guidelines that constitute a disciplined approach to software development. It is intended primarily for managers of software development efforts and for the technical personnel (software engineers, analysts, and programmers) who are responsible for implementing the recommended procedures. This document is neither a manual on applying the technologies described here nor a tutorial on monitoring a government contract. Instead, it describes the methodologies and tools that the SEL recommends for use in each life cycle phase to produce manageable, reliable, cost-effective software.”

### 4.4 Organization

The document begins with an overview of the classical waterfall software life cycle model. The remainder of the document is organized according to the phases of such a model: requirements definition, requirements analysis, preliminary design, detailed design, implementation, system testing, and acceptance testing. The last section contains ‘Dos’ and ‘Don’ts’ for project success that were based on research studies involving over 100 FDD projects.

### 4.5 Audience

According to the Introduction of the “third” revision, “This document…is intended primarily for managers of software development efforts and for the technical personnel (software engineers, analysts, and programmers) who are responsible for implementing the recommended procedures.” As noted above, readers outside the flight dynamics environment are cautioned that the SEL Recommended Approach may not necessarily be applicable to other environments.

### 4.6 Application Domain

The application domain is confined to flight dynamics projects at GSFC. Users of the document are cautioned that it may not be applicable to other environments.

### 4.7 Applicability to State of the Practice

The SEL Recommended Approach is significantly behind the current state of the practice, although it does accurately reflect the way software was being developed in the old Code 550, Flight Dynamics. Even the 1992 revision still reflects a mid-1980s approach to software development, relying on structured analysis, structured design, the classical waterfall model, etc. It should be noted that, as part of its program for ISO 9000 compliance, the ISC has recommended that the SEL Recommended Approach be revised for compatibility with ISO 9001. This would provide an opportunity to update the entire Recommended Approach to reflect the current state of the practice.

## 5. SSDM Standards and Procedures

### 5.1 History

The GSFC Mission Operations and Data Systems Directorate (MO&DSD) chose CSC as its prime contractor on the Systems, Engineering, and Analysis Support (SEAS) program to provide for the development of software systems from November 1987 to September 1997. In July 1989, CSC first published the SEAS System Development Methodology (SSDM) as a standard methodology for system development for MO&DSD. SSDM is based on Digital Systems Development Methodology (DSDM), a CSC proprietary software development methodology that dates from the late 1970s, but was heavily tailored to reflect the MO&DSD objectives and culture. It was also significantly influenced by the SEL Recommended Approach (see above) and CSC’s experience developing software for the old Code 550.

The SSDM Standards and Procedures (S&Ps), first published in June 1990, present detailed methods and techniques based on SSDM. Some of the standards contain templates, similar to military Data Item Descriptions (DIDs), for the format and contents of particular documents. Other standards provide templates for forms; still others describe how certain events (for example, a System Requirements Review) will be conducted. Procedures specify the steps in a process (for example, how to inspect and certify a unit design).

### 5.2 Scope

From the preface to SSDM: "SSDM…provides guidance on all aspects of system development, from concept development through system installation and acceptance, and addresses both hardware and software considerations." It addresses project management, systems engineering, and life cycle system development. Because it evolved from the earlier DSDM, which was confined to software, SSDM still retains a strong flavor of software development. SSDM does contain a chapter on hardware development, however, and the SSDM S&Ps contain three standards that are specific to hardware fabrication.

### 5.3 Purpose

The reason for adopting and following a standardized methodology such as SSDM is "to reduce the risks associated with the system development process." These risks can affect schedule, cost, functionality, and performance. "These problems occur periodically on the SEAS program, but they don't occur as often and the results are not as severe as in the past. That's because SSDM and its related standards and procedures reflect processes and practices that have been proven successful in developing systems."

### 5.4 Organization

The SSDM standards and procedures are organized into seven categories, as listed below. The numbers in parentheses indicate the number of standards and procedures contained in each section. The origins of SSDM in project management and software development are evident from the distribution of S&Ps among the categories.

Project Management (33)
Systems Engineering (3)
Hardware Development (3)
Software Development (24)
System Test and Evaluation (3)
Documentation (32)
Establishing Standards and Procedures (3)

### 5.5 Audience

There is no single audience for the entire collection of standards and procedures. The individual S&Ps have different audiences: project managers, system engineers, software developers, hardware developers, testers, CM personnel, QA personnel, operations personnel, and technical publications personnel.

### 5.6 Application Domain

The SSDM S&Ps were developed specifically to address the development of space data systems under the SEAS program for the Mission Operations and Data Systems Directorate (MO&DSD) at GSFC.

### 5.7 Applicability to State of the Practice

SSDM, like the SEL Recommended Approach, is significantly behind the current state of the practice. Although the SSDM S&Ps were formerly maintained in hard copy form, the entire set is now being maintained online. Nevertheless, even though nearly all the current online S&Ps are dated November, 1998, they differ in very few respects from the last hard-copy version, published in August, 1994. The online SSDM S&Ps as a whole still reflect a mid-1980s approach to software system development, relying on structured analysis, structured design, the waterfall life cycle model, a manual approach to CM, etc.

We should note that individual CSC organizations within SEAS (now called SETS) have implemented their own project-specific S&Ps that may reflect more recent technology. It would be worth reviewing these project-specific S&Ps to identify any that are significantly closer to the state of the practice than the official online set of SSDM S&Ps.

## 6. ISO 9000 Suite

### 6.1 History

In 1987, ISO Technical Committee 176, whose charter is quality, published the first version of the ISO 9000 suite of quality standards (ISO 9000 through 9004). A revised version of this suite was published in 1994. A second revision is being prepared for issuance in late 2000; this second revision is presently designated ISO Committee Draft #2 (CD2).

Almost as soon as the ISO 9000 standards were first published, the ISO software community began negotiating with the quality community to develop guidelines for applying ISO 9001 in a software environment. In 1990, ISO distributed the first draft of ISO 9000-3, "Guidelines for the application of ISO 9001 to the development, supply, and maintenance of software." There was concern, however, that the draft did not align with ISO 9001. This made it difficult to locate the software-specific guidance provided for any particular clause of ISO 9001. In 1997, therefore, an updated version of ISO 9000-3, aligned with ISO 9001:1994, was published. Another revision of ISO 9000-3 is planned for late 2000.

### 6.2 Scope

ISO 9001 addresses 20 requirements, including management responsibility, quality system, design control (which ISO 9000-3 interprets to cover software development), document and data control, training, statistical techniques, etc. The scope of CD2 has been greatly expanded, and organization has been substantially changed; see 6.4 below for the organization (and scope) of ISO 9001 CD2.

### 6.3 Purpose

ISO 9000 is a suite of standards for building, operating, and documenting a quality management system. ISO 9001 is a standard for quality in the design, development, production, installation, and servicing phases of the life cycle. ISO 9000-3 is a guide for the application of ISO 9001 to software.

### 6.4 Organization

ISO 9001:1994 is organized according to the 20 requirements alluded to in 6.2. ISO 9000-3:1997 follows the same organization. For reference, these 20 requirements are as follows:

1. Management responsibility
2. Quality system
3. Contract review
4. Design control
5. Document and data control
6. Purchasing
7. Control of customer-supplied product
8. Product identification and traceability
9. Process control
10. Inspection and testing
11. Control of inspection, measuring and test equipment
12. Inspection and test status

13. Control of nonconforming product
14. Corrective and preventive action
15. Handling, storage, packaging, preservation and delivery
16. Control of quality records
17. Internal quality audits
18. Training
19. Servicing
20. Statistical techniques

The committee draft for the upcoming revision of ISO 9001, designated ISO/CD2 9001:2000, has been completely reorganized. The organization of this committee draft (and of the new companion guide for performance improvement, ISO/CD2 9004:2000) follows a two-tier structure, as follows:

- Management Responsibility
  - General guidance
  - Interested party needs and expectations
  - Legal requirements
  - Policy
  - Planning
  - Quality management system
  - Management review
- Resource Management
  - General guidance
  - People
  - Information
  - Infrastructure
  - Work environment
  - Suppliers and partnerships
  - Natural resources
  - Finance
- Product and/or Service Realization
  - General guidance
  - Interested party related processes
  - Design and development
  - Purchasing
  - Production and service operations
  - Control of measuring and monitoring devices
- Measurement, Analysis and Improvement
  - General guidance
  - Measurement and monitoring
  - Control of nonconformity
  - Analysis of data for improvement
  - Improvement

Note that this organization for ISO 9001:2000 is likely to change as the current committee draft progresses through the life cycle of ISO documents. It is scheduled to be issued as a Draft International Standard (DIS) within the next several months, and then balloted and approved as an International Standard (IS) by the end of 2000.

### 6.5 Audience

The ISO 9000 suite of standards, including ISO 9001, was written by the ISO quality community, primarily for manufacturing environments. Its generic audience is the national bodies of the many countries that belong to ISO. It has been adopted by a variety of technical audiences, working in different application domains. The audience for ISO 9000-3:1997 is the international software engineering community, who are seeking guidance on the application of ISO 9001:1994 in software environments.

### 6.6 Application Domain

One of the reasons for the widespread interest in, and use of, the ISO 9000 suite of standards is that it was not written for any specific application domain. These standards are currently being used to certify projects and organizations in a wide range of application domains, including manufacturing, medicine, finance, transportation, and software.

### 6.7 Applicability to State of the Practice

The ISO 9000:1994 suite of standards, and also ISO/CD2 9001:2000, are reasonably close to the state of the practice. That having been said, it should be reiterated that these standards are not specific to software, but are designed to be applicable to any application domain. ISO 9000-3:1997, the guide for application of ISO 9001 to software environments, is reasonably close to the state of the practice. It is sufficiently generic that its applicability is not limited to any one paradigm for software development. An ISO effort is presently being initiated to prepare a guide for the application of ISO/CD2 9001:2000 to software.

## 7. Analysis

### 7.1 Similarities

Each of these five approaches represents a valid philosophy of software development. Except for ISO 9000, they all cover the entire scope of software engineering, from concept formulation through maintenance and operations. (ISO 9000 does not address the concept formulation or requirements definition phases; it begins with the system design phase.) The SEL Recommended Approach and the SSDM S&Ps are quite similar, both of them having been developed by CSC for the GSFC environment. Both of them also reflect a 1980s-era approach to software development. The IEEE SWE Standards, ISO/IEC/IEEE/EIA 12207, and ISO 9001 (with the guidance of ISO 9000-3) were prepared by, and for, larger and more dispersed technical communities. Any of these approaches is surely preferable to ad hoc, “seat of the pants” software development; any of them can be part of a strategy for developing software “cheaper, better, faster.”

### 7.2 Differences

Of the five approaches considered here, the IEEE SWE Standards probably present the best picture of the overall state of the practice for software engineering. I say this for four reasons:

- Each of the IEEE SWE standards is reviewed every five years for continued applicability
- Nearly all of the IEEE SWE Standards date from 1993 or later
- The IEEE SWE Standards cover the full gamut of software engineering, addressing requirements definition, design, coding, test, reuse, documentation, and maintenance
- The IEEE SWE Standards were written by knowledgeable experts in their respective fields and adopted by the IEEE through a rigorous balloting and consensus process.

It should, however, be noted that, although these standards are of consistently high quality, the IEEE SWE Standards Collection is not as comprehensive as it ought to be. Until quite recently, there has been no overall structure into which the standards fit, and no overall vision has driven the development of these standards. Someone would come forward and form a group to prepare a standard on a topic of current interest; that standard would eventually (in almost all cases) be reviewed, balloted, and ultimately adopted. The coverage is therefore spotty and incomplete.

ISO/IEC/IEEE/EIA 12207 currently has the widest user community of the five approaches. It, like the IEEE SWE standards, was prepared and balloted by knowledge experts in software life cycle processes. It also had the advantage that it did not require aggressive marketing, but was written by and for the community of software developers and acquirers, as a tool to be used to control the software development and acquisition processes.

ISO 9001 is taking hold fast; it is being used as the basis for certification of projects and organizations in a wide variety of application domains. Although technically written to address quality management, it employs an extremely broad definition of ‘quality’, and actually addresses most of the processes associated with life cycle system (or software) development. It should be noted, however, that the application of ISO 9001 ‘begins’ with system design; concept formulation and requirements definition are outside of its scope. It should also be noted that neither ISO 9001 nor the companion guide, ISO 9000-3, was ever intended to be used as a manual for system development. Its focus, rather, is on organizational certification for quality management. It thus approaches software development from a subtly different viewpoint from the other four approaches considered in this paper.

The SSDM S&Ps have the advantage of good coverage of the software engineering field. They provide a wealth of guidance for the total span of system and software development. They also reflect the influence of the SEL Recommended Approach, which encapsulates lessons learned from over 25 years of research in the Flight Dynamics environment. Certainly, there are holes; there is a list of perhaps 20 S&Ps that were planned but have never been written. But this deficiency is outweighed by the excellence and completeness of the individual standards, and by the consistency and integrity of the entire collection. The SSDM S&Ps were written to provide a comprehensive methodology for system and software development, and they perform that function extremely well. It can be argued that this strength outweighs the fact that many of the S&Ps are technically rather dated, and are not in synch with the current state of the practice.

The SEL Recommended Approach is quite similar to (and, as noted above, strongly influenced) SSDM. It is not as comprehensive as the SSDM S&Ps. It provides a valid approach to software development, but lacks the detailed guidance that effective S&Ps can provide. It is also, like the SSDM S&Ps and for the same reason, quite dated. It also suffers from its narrow development base: it is based on extensive research within the old GSFC Code 550, Flight Dynamics, an organization that no longer exists. Although many of the functions of the old Code 550 have been picked up by the present Code 583, it is not at all clear that the SEL Recommended Approach is appropriate for the Code 583 culture. One project manager in Code 584 at Wallops Island, however, reports that the software processes being used on her project were adopted from the SEL Recommended Approach.

### 7.3 Overlap in Coverage

The coverage of the respective approaches is depicted in Table 1.

### 7.4 Potential applicability to current ISC software work

The applicability of ISO 9001, 9000-3, and CD2 9001:2000 has been decreed by NASA. NASA Policy Directive (NPD) 2820.1 establishes consistency with ISO 9000-3 as a NASA-wide goal. Presumably this NPD will be updated to address ISO CD2 9001:2000. So quality management, with its implications for software engineering processes, is with us in the ISC environment and cannot be ignored.

The IEEE SWE Standards Collection, even with its holes, provides remarkably good coverage of the software engineering field, and presents a thorough and up-to-date set of standards. It is true that the standards suffer from a lack of consistency within the collection, but this weakness has been recognized and should be less evident in the next edition of the Collection. Although I suspect that the Standards Collection is probably not widely used within ISC, the software projects within ISC could probably benefit significantly from considered, selective adoption of some of the IEEE SWE Standards. They provide the elements of an effective, state-of-the-practice approach to software engineering.

**Table 1. Coverage of the software engineering domain.**

| Subject Area | IEEE SWE Standards | ISO/IEC 12207 | SEL Recommended Approach | SSDM S&Ps | ISO 9001, 9000-3, and CD2 9001:2000 |
|---|---|---|---|---|---|
| Program Management | X | X |  | X | X |
| S/W Life Cycle Processes | X | X | X |  |  |
| S/W Acquisition | X | X |  |  |  |
| Contract Preparation | X | X |  |  |  |
| Contract Review |  |  |  |  | X |
| System Engineering | X | X |  | X |  |
| System Requirements | X | X | X |  |  |
| Software Requirements | X | X | X | X |  |
| Software Design | X | X | X | X | X |
| Implementation | X | X | X | X | X |
| System Test | X | X | X | X | X |
| Acceptance Test |  | X | X | X |  |
| Operations | X | X |  | X |  |
| Maintenance | X | X |  | X |  |
| Quality Management |  |  |  |  | X |
| Quality Assurance | X | X | X | X | X |
| Configuration Management | X | X | X | X | X |
| Reuse | X |  | X |  |  |
| Documentation | X | X | X | X | X |
| Measurement | X |  | X |  | X |
| Training |  | X |  | X | X |

ISO/IEC 12207 is designed for management of the system acquisition process. ISC could certainly experiment with using ISO/IEC 12207 for the outsourcing of software development, maintenance, and operation contracts. It can also be helpful for suggesting activities to be undertaken when an organization within the ISC is developing or maintaining software for another GSFC code. ISC management should take note, however, that ISO/IEC does not constitute a manual or guidebook for software development, and should not be used as such. It is specifically intended to be used to control and manage the acquisition of software systems within an acquirer/supplier framework.

Outside of two projects in Code 584, it's not clear that anyone at ISC is still using the SEL Recommended Approach. This question should be addressed during the follow-on study that I have proposed. As noted in section 4.7, the planned ISC revision of the Recommended Approach for compliance with ISO 9001 may provide an opportunity to enlarge the scope of this document, fill in its holes, and make it consistent with the current state of the practice. This would increase the usability of the Recommended Approach throughout ISC.

The SSDM S&Ps have been extremely helpful to ISC and its predecessor organizations for the last ten years. Clearly, they have been found both useful and applicable to the type of software work performed with the application domains that comprise ISC. The underlying methodology, however, has become dated. This is one of the reasons that CSC has moved from the corresponding CSC methodology, DSDM, to the more up-to-date approaches represented by Catalyst and the current group of Civil Group processes. I believe that the applicability of the official SSDM S&Ps to the work done at ISC has already begun to decrease. This is no doubt why, as noted in section 5.7, individual CSC organizations within the SETS project have prepared tailored, project-specific S&Ps, based on the SSDM set. These tailored S&Ps may reflect more recent technology, and may better serve the needs of ISC than the official SSDM S&Ps. This question should be addressed during the follow-on study that I have proposed.

## 8. Summary of Research Study

This white paper is the first deliverable in a SEL research study. The object of this study is to analyze and compare the five SWE methodologies discussed above, and to investigate their impact thus far upon software efforts within the ISC. A second objective is to determine whether any or all of these standards and methodologies can be used, or are being used, together with current GSFC methodologies and standards.

The preparation of this white paper is the first step in the research study. The next step is to identify projects within the ISC that claim to be using some software methodology or set of standards. We then intend to interview the cognizant team leads or project managers to identify the precise standards and methodologies being followed.

In particular, we hope to identify any ISC software projects that use or have used IEEE standards, ISO 12207, ISO 9001, the SEL *Recommended Approach*, or *SSDM*. We also hope to capture the assessment of the team leads or project managers regarding the impact, if any, that the use of these standards and methodologies has had upon these projects.

As a subsequent follow-on study, the SEL could determine whether the use of IEEE, ISO, SEL, and CSC *SSDM* standards and methodologies, in concert with established GSFC/ISC standards and methodologies, can result in cheaper/better/faster software development within the ISC. This would necessitate setting up pilot projects to use these standards and methodologies, and control projects that would not use them. Such a study would probably require two years.

# GLOSSARY OF ACRONYMS

| | |
|---|---|
| ANSI | American National Standards Institute |
| CASE | Computer-Aided Software Engineering |
| CD2 | Committee Draft 2 |
| CM | Configuration Management |
| CSC | Computer Sciences Corporation |
| DIS | Draft International Standard |
| EIA | Electronic Industries Association |
| GSFC | Goddard Space Flight Center |
| IEC | International Electrotechnical Commission |
| IEEE | Institute of Electrical and Electronics Engineers |
| IS | International Standard |
| ISC | Information Systems Center |
| ISO | International Organization for Standardization |
| JTC1 | Joint Technical Committee 1 |
| MO&DSD | Mission Operations and Data Systems Directorate |
| QA | Quality Assurance |
| S&Ps | Standards and Procedures |
| SC7 | Subcommittee 7 |
| SEAS | Systems, Engineering, and Analysis Support |
| SEL | Software Engineering Laboratory |
| SLCP | Software Life Cycle Process |
| SSDM | SEAS Systems Development Methodology |
| SWE | Software Engineering |