SOFTWARE MANAGEMENT ENVIRONMENT (SME)

INSTALLATION GUIDE

JANUARY 1992

(NASA-TM-108578) SOFTWARE MANAGEMENT ENVIRONMENT (SME)
INSTALLATION GUIDE (NASA) 35 p
SOFTWARE MANAGEMENT ENVIRONMENT (SME)

INSTALLATION GUIDE

JANUARY 1992

NASA

National Aeronautics and Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771
The Software Engineering Laboratory (SEL) is an organization sponsored by the National Aeronautics and Space Administration/Goddard Space Flight Center (NASA/GSFC) and created to investigate the effectiveness of software engineering technologies when applied to the development of applications software. The SEL was created in 1976 and has three primary organizational members:

NASA/GSFC, Systems Development Branch
University of Maryland, Department of Computer Science
Computer Sciences Corporation, Systems Development Operation

The goals of the SEL are (1) to understand the software development process in the GSFC environment; (2) to measure the effect of various methodologies, tools, and models on this process; and (3) to identify and then to apply successful development practices. The activities, findings, and recommendations of the SEL are recorded in the Software Engineering Laboratory Series, a continuing series of reports that includes this document.

The major contributor to this document is

David Kistler (Computer Sciences Corporation)

Additionally, the following person contributed significantly:

Kellyann Jeletic (GSFC)

Single copies of this document can be obtained by writing to

Systems Development Branch
Code 552
Goddard Space Flight Center
Greenbelt, Maryland 20771
This document contains installation information for the Software Management Environment (SME), developed for the Systems Development Branch (Code 552) of the Flight Dynamics Division of Goddard Space Flight Center (GSFC). The SME provides an integrated set of management tools that can be used by software development managers in their day-to-day management and planning activities. This document provides a list of hardware and software requirements as well as detailed installation instructions and trouble-shooting information.
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Document Overview</td>
<td>1-1</td>
</tr>
<tr>
<td>2</td>
<td>Installation Procedure Overview</td>
<td>2-1</td>
</tr>
<tr>
<td>2.1</td>
<td>Installation Overview</td>
<td>2-1</td>
</tr>
<tr>
<td>2.2</td>
<td>Hardware Requirements</td>
<td>2-1</td>
</tr>
<tr>
<td>2.3</td>
<td>Software Requirements</td>
<td>2-2</td>
</tr>
<tr>
<td>2.4</td>
<td>Communications Requirements</td>
<td>2-2</td>
</tr>
<tr>
<td>2.5</td>
<td>Assumptions and Restrictions</td>
<td>2-2</td>
</tr>
<tr>
<td>3</td>
<td>SME File Description</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1</td>
<td>SME VAX Directory Structure</td>
<td>3-1</td>
</tr>
<tr>
<td>3.2</td>
<td>SME VAX Files</td>
<td>3-2</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Executable Images</td>
<td>3-2</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Compiled LISP Files</td>
<td>3-3</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Model Files</td>
<td>3-3</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Project Data Files</td>
<td>3-3</td>
</tr>
<tr>
<td>3.2.5</td>
<td>Definition Files</td>
<td>3-3</td>
</tr>
<tr>
<td>3.2.6</td>
<td>Command Files</td>
<td>3-3</td>
</tr>
<tr>
<td>3.3</td>
<td>SME PC Files</td>
<td>3-4</td>
</tr>
<tr>
<td>4</td>
<td>SME Installation and Configuration Procedures</td>
<td>4-1</td>
</tr>
<tr>
<td>4.1</td>
<td>VAX Installation Procedure</td>
<td>4-1</td>
</tr>
<tr>
<td>4.2</td>
<td>SME PC Installation Procedure</td>
<td>4-1</td>
</tr>
<tr>
<td>4.3</td>
<td>VAX Command File Configuration</td>
<td>4-2</td>
</tr>
<tr>
<td>5</td>
<td>Troubleshooting Hints</td>
<td>5-1</td>
</tr>
<tr>
<td>5.1</td>
<td>General Guidelines</td>
<td>5-1</td>
</tr>
<tr>
<td>5.2</td>
<td>Specific Anomalies</td>
<td>5-1</td>
</tr>
</tbody>
</table>
Table of Contents (Cont’d)

Appendix—Support File Reference List
Glossary
References
Standard Bibliography of SEL Literature
SECTION 1—DOCUMENT OVERVIEW

The Software Management Environment (SME) is being developed for the Systems Development Branch (Code 552) of the Flight Dynamics Division (FDD) of the Goddard Space Flight Center (GSFC). The SME provides an integrated set of management tools that can assist software development managers in their day-to-day management and planning activities.

This document contains the information you will need to install the SME software on a VAX computer connected to personal computer (PC) workstations. It is organized as follows:

- Section 2 presents an overview of the installation procedures and summarizes the minimum hardware, software, and communications requirements for the SME. General restrictions and assumptions are also summarized.
- Section 3 describes the SME directory structures as they exist on VAX computers and PC workstations, and presents a summary of the files and their contents.
- Section 4 describes how to install and configure the SME software.
- Section 5 provides troubleshooting hints for successful execution. General guidelines and specific anomalies are given.
- The appendix describes the SME configuration and support files required for execution.
SECTION 2—INSTALLATION PROCEDURE OVERVIEW

The following sections describe the hardware, software, and communications configuration you will need to install and run the SME on VAX computers and PC workstations. General assumptions and restrictions are also presented.

2.1 INSTALLATION OVERVIEW

Because of the distributed architecture of the SME, there are two stages to complete installation: loading the VAX-resident portion of the software, and loading the PC-resident portion of the software.

The VAX-resident portion of the software is distributed on a 6250 bits per inch (bpi) 9-track tape created using the virtual memory system (VMS) backup utility. This tape contains all files that you will need to execute the SME. After you unload the tape onto the target VAX computer, the files will appear in the directory structure that the SME expects (Section 3.1). Section 4.1 presents step-by-step procedures for installing the VAX-resident portion of the software.

The PC-resident portion of the SME software is distributed on a 360-kilobyte (kB) installation diskette that was formatted under Microsoft Disk Operation System (MS-DOS) Version 3.3. Section 4.2 presents complete instructions for installing the PC-resident portion of the SME software.

2.2 HARDWARE REQUIREMENTS

The VAX hardware and memory you will need to install and run the SME are as follows:

- VAX computer
- Asynchronous communications port
- 800-kB memory (peak working set size of 1500 pages)
- 3 megabytes (MB) minimum disk storage (5 MB optimum)

The PC hardware and memory you will need to install and run the SME:

- IBM PC or compatible
- Enhanced graphics adapter (EGA) color monitor
- Serial communication port (EIA RS-232C)

1 A video graphics array (VGA) color monitor may also be used for EGA graphics in VGA medium-resolution mode.
• 640-kB memory
• 125-kB disk storage

2.3 SOFTWARE REQUIREMENTS

The following lists the VAX and PC software you will need to install and run the SME:

• VAX—VMS operating system (Version 5.3 or higher)\(^1\)
• PC—MS-DOS (Version 3.0 or higher)

Additionally, VAX LISP must be available in order to run the SME’s expert system.

2.4 COMMUNICATIONS REQUIREMENTS

The following delineates the communications configuration required by the SME:

• Support for serial asynchronous communications
• Conformance to the EIA RS-232C communications standard
• Interface via a hard-wired connection (i.e., data terminal equipment (DTE) to DTE in a "null modem configuration") or via a modem (i.e., DTE to data communications equipment (DCE) to DTE)

2.5 ASSUMPTIONS AND RESTRICTIONS

The SME program assumes the SME user has appropriate READ, WRITE, and EXECUTE privileges for the directories created by the VAX SME software installation procedure. Additionally, the user must have a disk quota for creating files on the disk where the SME was installed.

The only installation-specific restriction concerns VAX user IDs. The maximum user ID length the current version of the SME will accept is 10 characters.

\(^{1}\)The SME requires runtime support for VAX system service routines and for VAX Pascal Version 4.0.
SECTION 3—SME FILE DESCRIPTION

The following sections provide an overview of the VAX and PC directory structures expected by the SME software. The files contained in the directories are also briefly described.

3.1 SME VAX DIRECTORY STRUCTURE

All files that the SME requires at runtime are distributed logically, based on usage, in a hierarchy of directories. The files needed by the SME include executable images, project data, models, and various definitions files. Figure 3-1 shows the directory structure that will be created by unloading the VAX SME software (Section 4.1) onto the target computer.

```
Figure 3-1. SME Directory Structure
```

The following is a list of the directories and a brief description of their contents. The numbers correspond to the numbers shown in Figure 3-1.

1. [SME] This directory contains the command files and executable images necessary to configure and run the SME software on the VAX. The directory also contains the [SME.DATA] directory. After execution of the SME, the [SME] directory will contain debug and user context files. As a result, users will require write privileges for this directory.
2. [SME.DATA] This directory contains the [KBASE], [LISP], [MODELS], [PROJECTS], and [USER] directories.

3. [SME.DATA.KBASE] This directory contains all knowledge base files that define explanations, factors, and reasons used by the trend assessment function under the SME growth monitor.

4. [SME.DATA.LISP] This directory contains all files required by the software to execute the VAX LISP expert system function under the SME rate monitor.

5. [SME.DATA.MODELS] This directory contains all models used by the SME at run time. The directory also contains files used to define the growth and profile measures used by the SME.

6. [SME.DATA.PROJECTS] This directory contains all the sample project data files provided on the installation tape.

7. [SME.DATA.USER] This directory contains the [SME.DATA.USER.DATA] directory and will contain all individual user directories created at runtime. Users will require write privileges for this directory.

8. [SME.DATA.USER.DATA] This directory contains all project-specific subjective data files created interactively by the user at runtime. These subjective data files are used with the knowledge base files for performing trend assessment under the SME growth monitor. Users will require write privileges for this directory.

9. [SME.DATA.USER.userx] This directory contains all project-specific phase estimate files created interactively by the user at runtime. The phase estimate files are used with the prediction function to record the basis of each prediction made by the user. Users will require write privileges for this directory.

3.2 SME VAX FILES

The SME requires several different types of files at program execution. The following sections describe these files. Additional information concerning the types of data used by the SME can be found in the Software Management Environment (SME) Concepts and Architecture (Reference 1).

3.2.1 Executable Images

The SME VAX software requires two executable images: SME_DRIVER.EXE and SME_EXEC.EXE. SME_DRIVER.EXE is simply a driver that enables the SME to execute either under a VAX VMS environment or under a VAX LISP environment. SME_EXEC.EXE functions as the executive-level driver for the system. Both executable images reside in the [SME] directory.
3.2.2 Compiled LISP Files

The SME VAX software requires several compiled VAX LISP files to run the expert system feature. These files reside in the [SME.DATA.LISP] directory. Note: the expert system is only available at those sites having VAX LISP available on the target VAX.

3.2.3 Model Files

The model files contain information used by the SME to create "guidelines" that, when compared to graphically plotted project data, can give the user an indication of how the project being plotted is behaving with respect to a "typical" project. The models are sequential text files read by the SME at runtime. All model files reside in [SME.DATA.MODELS] and have the file extension MDL.

3.2.4 Project Data Files

The sample project data files provided on the installation tape contain historical data used by the SME to generate plots that graphically represent a project's behavior over its development life cycle. The files are sequential text files that function as input to the SME and have the file extension DAT. All project data files reside in the directory [SME.DATA.PROJECTS]. A complete list of sample projects and available data for each project is provided with the distribution kit.

3.2.5 Definition Files

The SME reads a number of files at runtime to define key elements required by various SME functions. The files MEASURE.LST and PROFILE.LST are read at runtime to define the growth and profile measures used by the SME. These are sequential text files that reside in the directory [SME.DATA.MODELS]. Another file, DEFINITIONS.LST, resides in the [SME] directory and defines the functions used by the overall assessment feature.

Three files define the explanations, factors, and reasons, respectively, used by the knowledge base: EXPLAIN.KBS, FACTOR.KBS, and REASON.KBS. These are also sequential text files that reside in the directory [SME.DATA.KBASE].

One file, INTERPRETATIONS.DAT, resides in directory [SME.DATA.LISP] and contains text descriptions of mnemonic codes returned by the LISP expert system function. (This option is not available at sites that do not have VAX LISP.)

3.2.6 Command Files

There are three primary command files, SME.COM, SME_SETUP.COM, and SME_RUN.COM, all residing in the [SME] directory. SME_SETUP.COM defines a set of VAX logicals required by the SME at runtime, and SME_RUN.COM invokes the executable image. Both SME_SETUP.COM and SME_RUN.COM are invoked by SME.COM.
3.3 SME PC FILES

There are three EXE files in the SME directory on the PC: INSTALL.EXE, SME_CHAT.EXE, and PORTSTAT.EXE. INSTALL.EXE is the program provided on the diskette that performs the actual installation of the SME onto the hard drive of the target PC. SME_CHAT.EXE is the program that reads, interprets, and subsequently displays the information sent to the PC by the VAX. PORTSTAT.EXE is a utility program provided to help identify which PC communications port is attached to the VAX communications line. The PORTSTAT program’s use is further described in Section 4.2.

SMESTART.BAT is a batch file that establishes the PC graphics mode and invokes the PC software. This file is created by the INSTALL program, described in Section 4.2.

There are five additional support files used by the SME PC software at runtime: EGAVGA.BGI (a graphic device driver); and GOTH.CHR, LITT.CHR, SANS.CHR, and TRIP.CHR (graphic fonts).
The following sections describe the procedures used to install the SME software on target VAX and PC computers.

**4.1 VAX INSTALLATION PROCEDURE**

After mounting the SME installation tape on a 9-track 6250-bpi density tape drive, issue the following command (where MUA0 is the local tape drive name):

```
allocate MUA0:
```

VMS will confirm the successful allocation with a message similar to the following:

```
%DCL-I-ALLOC, _$1$MUA0: allocated
```

Next, issue the following command:

```
mount/foreign/density=6250 MUA0:
```

VMS will respond with a confirmation message similar to the following:

```
%MOUNT-I-MOUNTED, INSTAL mounted on _$1$MUA0: (STLH1)
```

Next, unload the tape using the VMS backup utility by issuing the following command:

```
backup MUA0:install.sme/save_set/select=[sme.test... ] -/noassist [target directory...]
```

The “target directory” is the directory into which you are unloading the SME installation tape. Be sure this directory specification includes the three trailing dots.

Finally, issue the following two commands to dismount and deallocate the tape:

```
dismount MUA0:
deallocate MUA0:
```

After installing the software from the tape, you can verify that all the files were unloaded successfully by referring to the listing of the tape creation log provided with the distribution kit.

**4.2 SME PC INSTALLATION PROCEDURE**

Insert the SME installation diskette into one of the PC disk drives. Change the default drive to reference the drive containing the diskette. For example, if you inserted the diskette in the B drive, enter the following at the DOS prompt:

```
b:
```

---

10000272
Before proceeding, you must know two facts about the particular PC on which you are installing the SME:

1. The number of the PC communications port attached to the VAX communications line (normally 1 or 2)

2. The baud rate of the communications line to the VAX (e.g., 4800 or 9600)

The definitive source for this information is the PC system coordinator for your organization. You can also use the SME PORTSTAT utility, included on the installation diskette, to assist in determining the appropriate number to reference the PC communications port. To run this utility, enter:

portstat

For each PC communications port found, the utility generates one of the two following messages:

- comX: status is HHHH
- comX: is not present

In these messages, X will be the port's number (i.e., 1 or 2), and HHHH will be the port's status displayed in hexadecimal.

If one communications port is reported as not present, you may assume the other port is attached to the VAX communications line. Some PCs, however, have multiple communications ports, and a status code will be reported for each port. In this case, consult the PC system coordinator to determine the proper port to reference.

Once you have determined the port number and baud rate for the VAX communications line, you can complete the installation by entering:

install

The INSTALL program will prompt you for the port number and baud rate before installing the needed SME PC software in the directory C:\SME on the hard disk.

4.3 VAX COMMAND FILE CONFIGURATION

Before you can execute the SME software successfully, you must resolve a few administrative and configuration issues.

First, you must provide all prospective SME users with VAX logon IDs, disk space allocation quotas, access to all SME files, and write privileges to the [SME], [SME.DATA.USER], and [SME.DATA.USER.DATA] directories.

Next, you must modify the command file SME_SETUP.COM that resides in the [SME] directory to reference the appropriate disk name correctly. Edit the file and change all
occurrences of "STL_DISK1" to the name of the disk on which you placed the SME files.

Finally, you must provide each user with a LOGIN.COM file that defines a logical "SME$SYSTEM" referencing your disk name concatenated with the [SME] directory. The LOGIN.COM should also define a VAX symbol "SME" that invokes the SME.COM file in SME$SYSTEM, so that the user can invoke the VAX portion of the software by entering:

SME

Then, if VAX LISP is available, the user will be able to invoke the SME with the expert system by entering:

SME EXPERT

Complete instructions for running the SME are provided in the Software Management Environment (SME) User Reference Material (Reference 1). A sample LOGIN.COM file is provided on the installation tape and resides in the [SME] directory as LOGIN_USER.COM.
The following sections provide some procedures to follow if you encounter problems when running the SME software.

5.1 GENERAL GUIDELINES

Once you have successfully installed the SME and tested all the options, you should create a backup tape. In the event that something catastrophic occurs and you have to reinstall the software, you will be installing your own configured version.

SME files can be edited, but with the exception of command (.COM) files, they should not be. Invalid characters, delimiters, or formats could inadvertently be introduced that would make further troubleshooting difficult if not impossible. If the files become hopelessly corrupted, you will probably have to reinstall the software.

If you are a first-time SME user, upon initialization you will receive the message that your context file could not be read and that default values are being used. This is not an error. The SME software creates a context file for you in the [SME] directory with a file name CONTEXT_userid.DAT. Assuming your context file does not get deleted, subsequent executions of the SME will not generate this message. Depending on the amount of use the SME receives, periodic purging of user context files is advised.

5.2 SPECIFIC ANOMALIES

The following list describes potential problems and corrective actions.

1. Problem: SME fails to execute.
   Response: Ensure that you are in the [SME] root directory and that SME_DRIVER.EXE and SME_EXEC.EXE are present.
   If you entered the “expert” parameter when attempting to run the software, ensure that LISP is present. If LISP is not present, reexecute the SME without the “expert” parameter.

2. Problem: SME fails to initialize properly (executive menu never appears).
   Response: Refer to the table in the appendix to check the logical references in SME_SETUP.COM; reconfigure, if necessary.
   Ensure MEASURE.LST and PROFILE.LST files are in the directory referenced by the “USRDATA” logical.
3. Problem: SME appears to “hang” when entering the PREDICT option of the growth monitor.

Response: Ensure that your userid has write privileges for [SME.DAT.USER.userid] directory.

Ensure that your userid has a quota for creating files and that the quota has not been exceeded.

Ensure that the logical “usrdta” in SME_SETUP.COM is not concatenated with a second logical containing a trailing dot.

4. Problem: SME appears to “hang” when attempting to change a reason in the ASSESS option of the growth monitor.

Response: See response 3.

5. Problem: SME appears to “hang” when entering the OVERALL ASSESSMENT option of the executive menu.

Response: Ensure that the DEFINITIONS.LST file is in the [SME] root directory.

6. Problem: SME generates “Unable to write context file” message upon exit.

Response: Ensure that your userid has a quota for creating files and that the quota has not been exceeded.

7. Problem: SME appears to “hang” when generating a graph.

Response: Suspect corrupted data. See if any model or data file for the project and/or measure has a different version number (higher) or create date (later) than the other model or data files. If this is the case and the original file still exists, delete the newer file. If the original file does not exist (it may have been purged), restore the original file from a backup tape. It might be possible to reconstruct the file by hand, but it is highly unlikely.
APPENDIX—SUPPORT FILE REFERENCE LIST

The SME requires that five logical references be defined at runtime, which is performed by SME_SETUP.COM. The following listing presents the "logicals" and the directory referenced by each logical after installation.

<table>
<thead>
<tr>
<th>Logical</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRJDATA</td>
<td>diskname:[SME.DATA.PROJECTS]</td>
</tr>
<tr>
<td>MDLDATA</td>
<td>diskname:[SME.DATA.MODELS]</td>
</tr>
<tr>
<td>KBSDATA</td>
<td>diskname:[SME.DATA.KBASE]</td>
</tr>
<tr>
<td>LSPDATA</td>
<td>diskname:[SME.DATA.LISP]</td>
</tr>
<tr>
<td>USRDATA</td>
<td>diskname:[SME.DATA.USER]</td>
</tr>
</tbody>
</table>

You may redefine the logicals to reference different directories by editing SME_SETUP.COM. The SME, however, expects the contents of the directories referenced by the logicals to be what is described in Section 3. For example, the models, MEASURE.LST, and PROFILE.LST files must reside in whatever directory is referenced by the logical MDLDATA.

The following listing presents what the SME expects in the directories referenced by each logical. It is a summary of the information contained in Section 3 and is included as a convenience.

<table>
<thead>
<tr>
<th>Logical</th>
<th>Expected Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRJDATA</td>
<td>sample project data supplied with the installation tape</td>
</tr>
<tr>
<td>MDLDATA</td>
<td>all models, MEASURE.LST, PROFILE.LST</td>
</tr>
<tr>
<td>KBSDATA</td>
<td>EXPLAIN.KBS, FACTOR.KBS, REASON.KBS</td>
</tr>
<tr>
<td>LSPDATA</td>
<td>compiled (FAS) LISP files, INTERPRETATIONS.DAT</td>
</tr>
<tr>
<td>USRDATA</td>
<td>DATA.DIR, user.DIR</td>
</tr>
</tbody>
</table>

Refer to Section 4.3 for more information on modifying the SME_SETUP.COM file.
GLOSSARY

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>bpi</td>
<td>bits per inch</td>
</tr>
<tr>
<td>DCE</td>
<td>data communications equipment</td>
</tr>
<tr>
<td>DTE</td>
<td>data terminal equipment</td>
</tr>
<tr>
<td>EGA</td>
<td>enhanced graphics adapter</td>
</tr>
<tr>
<td>FDD</td>
<td>Flight Dynamics Division</td>
</tr>
<tr>
<td>GSFC</td>
<td>Goddard Space Flight Center</td>
</tr>
<tr>
<td>kB</td>
<td>kilobyte</td>
</tr>
<tr>
<td>MB</td>
<td>megabyte</td>
</tr>
<tr>
<td>MS-DOS</td>
<td>Microsoft Disk Operating System</td>
</tr>
<tr>
<td>PC</td>
<td>personal computer</td>
</tr>
<tr>
<td>SME</td>
<td>Software Management Environment</td>
</tr>
<tr>
<td>VGA</td>
<td>video graphics array</td>
</tr>
<tr>
<td>VMS</td>
<td>virtual memory system</td>
</tr>
</tbody>
</table>
REFERENCES


STANDARD BIBLIOGRAPHY OF SEL LITERATURE

The technical papers, memorandums, and documents listed in this bibliography are organized into two groups. The first group is composed of documents issued by the Software Engineering Laboratory (SEL) during its research and development activities. The second group includes materials that were published elsewhere but pertain to SEL activities.

SEL-ORIGINATED DOCUMENTS

SEL-76-001, Proceedings From the First Summer Software Engineering Workshop, August 1976

SEL-77-002, Proceedings From the Second Summer Software Engineering Workshop, September 1977

SEL-77-004, A Demonstration of AXES for NAVPAK, M. Hamilton and S. Zeldin, September 1977

SEL-77-005, GSFC NAVPAK Design Specifications Languages Study, P. A. Scheffer and C. E. Velez, October 1977

SEL-78-005, Proceedings From the Third Summer Software Engineering Workshop, September 1978


SEL-78-007, Applicability of the Rayleigh Curve to the SEL Environment, T. E. Mapp, December 1978


SEL-79-005, Proceedings From the Fourth Summer Software Engineering Workshop, November 1979


SEL-80-005, *A Study of the Musa Reliability Model*, A. M. Miller, November 1980

SEL-80-006, *Proceedings From the Fifth Annual Software Engineering Workshop*, November 1980


SEL-81-012, *The Rayleigh Curve as a Model for Effort Distribution Over the Life of Medium Scale Software Systems*, G. O. Picasso, December 1981


BI-2


SEL-82-102, *FORTRAN Static Source Code Analyzer Program (SAP) System Description (Revision 1)*, W. A. Taylor and W. J. Decker, April 1985

SEL-82-105, *Glossary of Software Engineering Laboratory Terms*, T. A. Babst, M. G. Rohleder, and F. E. McGarry, October 1983


SEL-87-008, *Data Collection Procedures for the Rehosted SEL Database*, G. Heller, October 1987


BI-4
SEL-88-002, Collected Software Engineering Papers: Volume VI, November 1988
SEL-88-004, Proceedings of the Thirteenth Annual Software Engineering Workshop, November 1988
SEL-89-005, Lessons Learned in the Transition to Ada From FORTRAN at NASA/Goddard, C. Brophy, November 1989
SEL-89-007, Proceedings of the Fourteenth Annual Software Engineering Workshop, November 1989
SEL-90-002, The Cleanroom Case Study in the Software Engineering Laboratory: Project Description and Early Analysis, S. Green et al., March 1990
SEL-90-004, Gamma Ray Observatory Dynamics Simulator in Ada (GRODY) Experiment Summary, T. McDermott and M. Stark, September 1990
SEL-90-005, Collected Software Engineering Papers: Volume VIII, November 1990


SEL-91-102, *Software Engineering Laboratory (SEL) Data and Information Policy (Revision 1)*, E. McGarry, August 1991


**SEL-RELATED LITERATURE**


BI-7


BI-9


National Aeronautics and Space Administration (NASA), NASA Software Research Technology Workshop (Proceedings), March 1980


6Seidewitz, E., “General Object-Oriented Software Development with Ada: A Life

9Seidewitz, E., “Object-Oriented Programming Through Type Extension in Ada 9X,”
Ada Letters, March/April 1991

4Seidewitz, E., and M. Stark, “Towards a General Object-Oriented Software Develop-
ment Methodology,” Proceedings of the First International Symposium on Ada for the
NASA Space Station, June 1986

9Seidewitz, E., and M. Stark, “An Object-Oriented Approach to Parameterized Soft-

Seventh Washington Ada Symposium, June 1990

7Stark, M. E. and E. W. Booth, “Using Ada to Maximize Verbatim Software Reuse,”

5Stark, M., and E. Seidewitz, “Towards a General Object-Oriented Ada Lifecycle,”
Proceedings of the Joint Ada Conference, March 1987

Ada,” Proceedings of the Tenth International Conference of the Chilean Computer Science
Society, July 1990

7Sunazuka, T., and V. R. Basili, Integrating Automated Support for a Software Manage-
ment Cycle Into the TAME System, University of Maryland, Technical Report TR-2289,
July 1989

Turner, C., and G. Caron, A Comparison of RADC and NASA/SEL Software Develop-
ment Data, Data and Analysis Center for Software, Special Publication, May 1981

Turner, C., G. Caron, and G. Brement, NASA/SEL Data Compendium, Data and
Analysis Center for Software, Special Publication, April 1981

Valett, J. D., and F. E. McGarry, “A Summary of Software Measurement Experiences
in the Software Engineering Laboratory,” Proceedings of the 21st Annual Hawaii
International Conference on System Sciences, January 1988

Weiss, D. M., and V. R. Basili, “Evaluating Software Development by Analysis of
Changes: Some Data From the Software Engineering Laboratory,” IEEE Transactions
on Software Engineering, February 1985

5Wu, L., V. R. Basili, and K. Reed, “A Structure Coverage Tool for Ada Software Sys-

1Zelkowitz, M. V., “Resource Estimation for Medium-Scale Software Projects,” Pro-


NOTES:

1This article also appears in SEL-82-004, *Collected Software Engineering Papers: Volume I*, July 1982.

2This article also appears in SEL-83-003, *Collected Software Engineering Papers: Volume II*, November 1983.


4This article also appears in SEL-86-004, *Collected Software Engineering Papers: Volume IV*, November 1986.

5This article also appears in SEL-87-009, *Collected Software Engineering Papers: Volume V*, November 1987.

6This article also appears in SEL-88-002, *Collected Software Engineering Papers: Volume VI*, November 1988.


8This article also appears in SEL-90-005, *Collected Software Engineering Papers: Volume VIII*, November 1990.

9This article also appears in SEL-91-005, *Collected Software Engineering Papers: Volume IX*, November 1991.