SAE The Engineering Society For Advancing Mobility Land Sea Air and Space ®
INTERNATIONAL
400 Commonwealth Drive, Warrendale, PA 15096-0001

SURFACE VEHICLE/ AEROSPACE STANDARD

SAE JA1004

ISSUED JUL1998

Issued 1998-07

An American National Standard

Software Supportability Program Standard

Foreword—In 1994 the SAE G-11 Reliability, Maintainability, Supportability, and Logistics (RMSL) Division chartered a software committee, G-11SW, to create several software standards and guidance documents across the RMSL spectrum, including a software supportability program standard. The committee was formed as a cross section of international representatives from commercial industries and governments.

The G-11SW committee has attempted to develop a standard that is consistent with a SAE G-11 system level supportability program standard and augmented by necessary software-specific support information. The G-11SW committee believes this document reflects the best current commercial practices, and meets the objectives of the United States Department of Defense Acquisition Reform initiative. This document is performance based and is intended to be used by industries to address market demands for supportable software products that facilitate system evolution, time to market, and implementation of cost-effective functionality. As appropriate, governments may also reference this document.

Software has been recognized by SAE G-11 as an important system component that is not adequately addressed at the system level. Software requires interpretation and variations on RMSL methods used by hardware. This document relies on the simple concept of supplier-customer dialogue and partnership to define, meet, and demonstrate assurance of software product supportability requirements. This document describes, within a Plan-Case framework, what performance requirements are necessary. An accompanying implementation guide (SAE JA1005) sets forth current best practices for how to structure the Plan in terms of activities, tasks, and methods so as to achieve the requirements of this document and provide demonstration evidence of supportability achievement in the form of a Case.

Development of this document and the accompanying implementation guide, as well as other supporting information reports, has required dedication by a few participants and extended review by a wider audience of potential users. The professionalism of all these individuals and the support they received from their companies, governments, and other organizations is gratefully acknowledged.

Abstract—This SAE Standard defines recommended practices for the achievement of suitable supportability and through-life support arrangements for software within an overall systems engineering framework. The Software Supportability Plan (goals to achieve) and the Software Supportability Case (demonstration of achievement) are presented as the basis for program management. This document is applicable to all projects incorporating software, and aims to meet the needs of end-users and of organizations that acquire, develop, or provide post-delivery support for software.

SAE Technical Standards Board Rules provide that: “This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user.”

SAE reviews each technical report at least every five years at which time it may be reaffirmed, revised, or cancelled. SAE invites your written comments and suggestions.

TO PLACE A DOCUMENT ORDER; (724) 776-4970 FAX: (724) 776-0790
SAE WEB ADDRESS http://www.sae.org

Printed in U.S.A.

TABLE OF CONTENTS

1. Scope—This SAE Standard defines the basic structural elements, and guidance on compilation and management, for a software supportability program. Software supportability considerations include initial design influence and through-life support embracing the operational use, post-delivery modification, and logistics management of software. This document requires that the processes of design, development, selection, and production of software include software supportability considerations, as relevant to particular project needs.

1.1 Field of Application—This document generally applies to all types of computer-based systems and throughout the project life-cycle. The developmental scope of the project and other issues as covered within the document determine how this document needs to be tailored.

1.2 Relationship to Project Management Plans—The concepts put forward in this document should be seen as extensions to existing project management practice, and may be integrated within other project management mechanisms provided that supportability progress is identifiable and traceable. When specific software supportability programs are initiated, they should be viewed as an integral part of the overall planning and control activity for a system project. This is essential in order to ensure that the cost and effort put into pure software aspects is at the appropriate level to achieve an optimized system support strategy.

1.3 Associated Guidance—General background information on software supportability is provided in [SAE AIR5121], which provides a top-level context in which to view the through-life support requirements for software, and concepts for how these needs might be addressed. A description of software support concept in terms of software support profiles, functional support areas, and support classes is contained in SAE JA1006. Guidance on implementation issues for a Software Supportability Plan and Software Supportability Case including the activities, methods, and tools concerned with a software supportability program is given in SAE JA1005. Other SAE program standards, such as [JASUPP] on system supportability, may be consulted as appropriate. Other government standards and guidance can be found in references [DEFSTAN0060], [MILHDBK502], and [MILPRF49506].

2. References

2.1 Applicable Publications—The following publications form a part of the specification to the extent specified herein. Unless otherwise indicated the latest revision of SAE publications shall apply.

2.1.1 SAE PUBLICATIONS—Available from SAE, 400 Commonwealth Drive, Warrendale, PA 15096-0001.

SAE AIR 5121—Software Supportability—An Overview
SAE JA1005 (Draft)—Software Supportability Implementation Guide
SAE JA1006 (Draft)—Software Support Concept
SAE JASUPP (Draft)—Supportability Program Standard

2.1.2 MILITARY STANDARDS—Available from DODSSP, Subscription Services Desk, Building 4D, 700 Robins Avenue, Philadelphia, PA 19111-5094.

MIL-HDBK-502—DoD Handbook Acquisition Logistics— May 30, 1997
MIL-PRF-49506—Performance Specification Logistics Management Information—November 11, 1996

2.1.3 UK MILITARY STANDARD—Available from Directorate of Standardization, Room 1138, Kentigern House, 65 Brown Street, Glasgow UK, G2 8 EX.

DEF STAN 00-60—"Integrated Logistic Support", Issue 2, March 1998. “Logistic Support Analysis Application to Software Aspects of Systems”, Part 3

3. Definitions

3.1 Software Support—The set of activities necessary to ensure that an operational software system or component fulfills its original requirements and any subsequent modifications to those requirements. Activities include all processes, resources, and infrastructure required in order to provide support throughout the software’s operational life.

3.2 Software Supportability—A set of attributes of software design, the associated development tools and methods, and the support environment infrastructure that enable the software support activities to be accomplished.

3.3 **Software Modification Support—**The software support activities of change analysis, implementation, test, and release of software products. Changes may be termed corrective, perfective and adaptive, and may also embrace modifications that are designed to prevent foreseeable future software operating problems.

3.4 **Software Operational Support—**The software support activities related to the day-to-day operation of software by the software user. Activities may include installation, configuration, data preparation and loading/unloading, backup, recovery, failure reporting, and training.

3.5 **Software Logistics Management Support—**The software support activities related to the "bridge" between the software support services for operational use and for post-delivery modification support. Activities may include help desk management, problem reporting and corrective action coordination, distribution of releases to field sites, management of configuration information concerning software releases in the field, and network communications among field sites and modification support sites.

3.6 **Software Maintainability—**The ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment. Also, a set of attributes that bear on the effort needed to make specified modifications.

3.7 **Software Maintenance—**The process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment.

NOTE—Software maintenance as defined, is essentially the same as software modification support. Software maintenance and the maintenance process have not always been interpreted as defined in this document. Hence, this document uses the term Modification Support to more clearly distinguish and contrast with the Operational and Logistics Management Support activities.

4. Software Supportability Program

4.1 **Objectives—**The objectives of a software supportability program are two-fold:

a. Ensure the delivery of a software product that meets the desired supportability characteristics; and
b. Ensure the support service provided for a deliverable software product meets the customer's needs and is balanced for cost, quality, and responsiveness.

4.2 **Principles—**In creating and conducting a software supportability program, there are three core principles that suppliers shall follow. These principles are that customer requirements for software supportability shall be:

a. Determined;
b. Met; and
c. Demonstrated.

These principles are the foundation framework for this document. Their application is based on the use of the twin artifacts of a Software Supportability Plan and Software Supportability Case, as described in the following sections. The Plan and Case are general purpose management tools that are suitable for use in many fields of system engineering and will be referenced throughout the suite of Society of Automotive Engineers (SAE) Reliability, Maintainability, Supportability, and Logistics program standards.

5. ***Management Requirements*—Plan and Case Framework**—The framework for management of software supportability is built around two key components: the Software Supportability Plan and the Software Supportability Case.

a. **Software Supportability Plan**—The Plan addresses software aspects of the System Supportability Plan, and describes the activities that are to be undertaken to achieve software supportability objectives. The Plan also describes activities that are to be undertaken to demonstrate that software supportability objectives have been achieved.

b. **Software Supportability Case**—The Case provides a justification of the approach and documents evidence which verifies that the software meets its supportability requirements.

5.1 Roles of the Software Supportability Plan and Case—The Software Supportability Plan and Case may be seen as having variable purposes depending on whether they are used as contractual, developmental, or support mechanisms. In the contractual context, the plan and case can be used as formal deliverables to be jointly defined and approved between a customer and supplier. The Plan and Case can contribute to any prospective development and may be requested during contract negotiations. Further guidance on contractual application is provided in Section 7 of this document. In a developmental context, the Plan and Case may serve the internal needs of a development organization in meeting supportability objectives. Finally, for organizations involved in providing post-delivery support for software, the Plan and Case may be used to establish or validate cost and efficiency objectives for software support processes, including for existing Off The Shelf (OTS) software.

5.2 Issues Affecting Application Strategy

a. **Product Complexity**—Software projects often involve the delivery of products comprising multiple, integrated components. Distinct software supportability programs may not be justified for all elements within such a structure. The key issue will be whether a particular software component is a separate contractual deliverable. If this is not the case - for instance, when software is included as an 'embedded' item within a deliverable equipment - the supportability plan for that equipment or relevant higher assembly should include any specific considerations for software. However, the requirements and guidance given in this document should nevertheless be applied.

b. **Custom Versus COTS Software**—Software may be delivered to a customer in a variety of product forms, ranging from new custom developments, through variant or partially modified items, to re-used or COTS packages. The concept of a software supportability program is valid across this range. However, the practical approach may be tailored according to the extent to which design influence is feasible, and in the light of any software support resources or infrastructure which are already in place.

5.3 Managing Hardware/Software Dependencies—Active management of hardware/software dependencies should be undertaken throughout the product life-cycle, as part of an overall systems engineering philosophy. During both initial development and subsequent post-delivery support, design changes may be proposed to hardware which have an effect on software, and vice-versa. The trade-off studies conducted in such cases should consider not only direct operational aspects, but also the implications to overall system supportability and through-life support costs.

6. Technical Requirements

6.1 Determining Customer Requirements

6.1.1 GENERAL—It is important for suppliers to understand their customers' business and system environments, within which a required software product will operate. However, the approach to gaining this understanding will depend on the nature of the product and other commercial factors.

Where a supplier is responding to a custom software development requirement, the particular supportability needs of the customer are of prime importance. The customer should describe supportability concepts and objectives and formal arrangements (e.g., contract, contract deliverables) may be established to facilitate customer/supplier dialogue on supportability matters.

Where a supplier's main business is the development and marketing of general-purpose software, the definition of supportability requirements for a new product will be approached somewhat differently. In this case, the prime focus will be the supplier's own assessment of the factors that characterize the expected customer base. However, there will be a common underlying objective, which is to provide high quality products and support services that promote customer satisfaction and supplier profitability.

6.1.2 FACTORS TO BE CONSIDERED—The activity of determining customer requirements for software supportability should be coordinated with equivalent activity in other RMSL disciplines, and as part of an overall systems approach. This activity, particularly for new product development, may require exploratory studies and some subsequent iteration or evolution to obtain an adequately robust requirements definition at an acceptable cost. An illustrative range of factors for consideration is outlined in the following paragraphs.

6.1.2.1 Conditions of Use—Supportability concerns depend on the operational use of the software. Some of these concerns can be determined from:

a. Business/mission profiles that the software is required to enable;
b. Operating environment for software users (e.g., air/land vehicle, office, in the field);
c. Functions to be provided by software that are critical to business/mission success;
d. The required availability and induced loss (economic or other) of the software not working as required;
e. Functional interaction with other software/systems;
f. Required in-use life of the software;
g. Anticipated business/mission requirements change profile over the life of the software;
h. Level of customer's experience or training requirements as a potential user; and
i. Special considerations such as security and maintenance personnel safety.

6.1.2.2 Supportability Characteristics—The customer may have standard requirements for characteristics of the software product, the processes used to develop and support the software, and the environment in which the software is to be developed and supported. Such characteristics may have primary objectives other than supportability, but they will invariably have a supportability or support cost impact and should therefore be identified and assessed within the software supportability program.

6.1.2.3 Support and Service Levels—The level of support for implementing software changes and providing customer service depends on several factors including:

a. Number, complexity, and priority of support requests;
b. Response times to failure incidents;
c. System recovery times following a failure;
d. Software and data loading/unloading times;
e. Response criteria for customer change requests;
f. Criteria for help/query services;
g. Customer's anticipated in-house software support capability;
h. Support center location/access criteria;
i. Customer's strategy for through-life technology refreshment; and
j. Constraints on supportability and support processes (e.g., need for safety/security certification; time/calendar periods when users will allow release of the system for planned maintenance or installation of upgrades).

6.2 Meeting Customer Requirements

6.2.1 GENERAL—There are two aspects to meeting a customer's software supportability requirements. The first is ensuring the delivery of a product with the appropriate design characteristics to facilitate the expected demand for through-life change and enhancement. The second aspect is the provision of a support capability that satisfies the customer's quality of service needs at an acceptable cost. These are interrelated goals that should be addressed through a coordinated approach to software supportability planning.

Software supportability planning may be conducted as a separately managed task, or as a sub-element of a broader planning activity for system development or support. The latter approach may be appropriate where a customer makes no specific software supportability requirements, or where requirements are fully satisfied by existing corporate standards and codes of practice. In the guidelines that follow, the term "plan" should therefore be interpreted as a general concept that may be implemented in various ways according to the nature of a particular project.

6.2.2 THE SOFTWARE SUPPORTABILITY PLAN—The software supportability plan is the means through which activities and progress in satisfying customer supportability requirements are controlled and monitored. The plan should provide clear traceability to original customer requirements for supportability, and should also show activities relating to the generation of the associated Software Supportability Case. The plan should be traceable to the broader planning activity for both system supportability and the acquisition/delivery of support arrangements for the overall fielded system. The plan should also be integrated appropriately with relevant software development and quality planning. The plan should be mutually agreed between the customer and the supplier before implementation, and subjected to appropriate management reviews during its period of use.

6.2.2.1 *Scope and Tailoring*—The main scope of a Software Supportability Plan should cover both designing for supportability and the provision of any resources or infrastructure necessary to satisfy customer requirements or assessed needs for support services. In practice, an organization might address these aspects separately in development and logistics planning areas, but where this occurs, the information and coordination links should be clearly defined and actively managed.

The format and content of the plans for a particular project will depend on the extent to which design influence (on supportability features) is feasible, and on the unique characteristics of the product to be delivered. If a project is developmental, the full range of elements listed as follows may be covered. However, if a project involves the delivery of unaltered OTS software, the main focus of the Plan will be on support services. The implementation guide [SAE JA1005] will cover in more detail possible structure and content of the software supportability plan.

6.2.2.2 *Design Aspects*—This part of a software supportability plan will deal with the design and development activities that are needed to determine how to adequately satisfy the customer's software supportability requirements. Typical aspects covered within a plan will include the following:

a. Supportability Analysis—Activities/tasks that provide an understanding of the desirable characteristics of software product, processes, and environment that enhance the supportability of the operational system. The following aspects may be included:

1. Assessments of support tools and techniques, test environments, configuration management tools, interoperability requirements between the support and operational environments;
2. Analysis of existing/planned development and support processes for configuration management, quality engineering, and change processing; transition of processes from development to support; and,
3. Identification of design standards and characteristics of the software product (e.g., documentation, source code, test scripts) that affect supportability.

b. Support Analysis—Activities/tasks that provide definition for the software support functions that will be provided. The following aspects may be included:

1. Estimates of through-life change demands on the software;
2. Resource allocation (including planned customer-provided resources, customer-configurable functions), duration, and frequency for each support function;
3. Implementation of functions in hardware or software;
4. Functionality, cost, support trade-offs when integrating OTS software; and
5. Supportability impacts on program design milestones.

6.2.2.3 Support Aspects—This part of a Software Supportability Plan will cover the activities concerned with the definition and acquisition of the necessary post-delivery support services for the customer. There are three distinct elements to software support that might be included:

a. Operational Use—Activities/resources concerned with sustaining operational use of the software in the field, either by the customer or by other agreed resources. Typical aspects to be covered may include:

1. Mission support equipment, support software, peripherals, storage space, interfacing equipment for pre- and post-mission support, and backup/recovery equipment required to sustain full operational capability of the software;
2. Number and type of specialist staff requirements for the installation, management, and operation of the software, including any field user training needs;
3. Additional products (User Manuals, License codes, Training Courses, Diagnostic Tools, etc.) delivered to the customer to support operational use of the software; and
4. Procedures for conducting activities in the field such as pre- and post-mission data load/collection, backup/recovery, failure reporting, and training.

b. Logistics Management—Activities/resources that represent the ‘bridge’ between the support services for operational use and for post-delivery modifications. The following minimum aspects should be covered:

1. Procedures and communications mechanisms to enable delivery of the required support services, updated software products, and interface with the user; procedural aspects defining the transition of the delivered product to field units may be included;
2. Physical media on which the software will be stored and delivered to the customer, as well as the packaging or other special (e.g., security) requirements; and
3. Procedures, data, and reporting mechanisms to ensure that hardware with software/firmware configurations are properly documented, tracked and managed, so that compatibility will be maintained in the fielded system throughout its operational life.

c. Post-Delivery Modifications—Activities/resources concerned with implementing required changes and improvements to the design configuration of a software product after initial delivery. The following aspects may be included:

1. Buildings and associated services to house the support environment and human resources;
2. Host machines, databases, software tools, and test equipment to be used for software support and change development;
3. Number and type of specialist staff requirements for software change management and production, and for customer help services; any training needs may also be covered; and
4. Change analysis, implementation, test, and release engineering processes along with software configuration management and quality engineering processes; transition between development and support processes including transfer of engineering information and methods.

6.2.3 DOCUMENTING SUPPORT ARRANGEMENTS FOR DELIVERED SOFTWARE—Software supportability planning should normally include the production of suitable documentation to describe the support arrangements put into place for a fielded software product. The scope of such documentation, and the extent to which it is formally delivered to the customer, will depend on the nature of the product and how it is procured or marketed.

For general-purpose commercial software, customers may only need visibility of the available infrastructure aspects for product support and user help. Other legal documentation for Intellectual Property Rights to software may be necessary to minimize risk due to supplier business failure, failure to deliver required support services, or change in business products supported. Customers for major custom software systems may require a more detailed description of support services. The supplier may internally maintain a complete definition of the facilities, resources, and infrastructure required to support a particular software product for each specific customer.

6.3 Demonstrating Requirements Satisfaction

6.3.1 THE SOFTWARE SUPPORTABILITY CASE—This document uses the core concept of the Software Supportability Case as the means by which suppliers can demonstrate that customer requirements for software supportability have been achieved in a particular product. A Software Supportability Case should be a readable overview of the evidence, including references to more detailed evidence as appropriate, to support requirements satisfaction. Various types of information as outlined as follows are possible. A Case document may be used both for the acceptance or assurance of new products and for substantiating supportability claims to potential new customers of proposed and existing products.

A Software Supportability Case should be compiled as design and development decisions are made. The approach to be applied for a particular product should be defined at the outset of supportability planning, and implemented across the life-cycle through a suitable activity schedule.

6.3.1.1 Scope—The scope of a Software Supportability Case should cover characteristics of both the deliverable product and the process (i.e., the environment of tools, methods, procedures, personnel) through which the product was developed and fielded, and will subsequently be operated and supported.

6.3.1.2 Structure and Content of The Case

a. Supportability Requirements—A Software Supportability Case will require clear traceability to the original project supportability requirements, elicited as described in 6.1. Those requirements that are to be addressed in the Case should be listed and categorized. The main topic breakdown should cover characteristics related to the software product, the development process, and the overall environment of support arrangements. Where appropriate, amplifying information on rationale for supportability requirements, or associated models/scenarios on which they are based, may be included in appendices to the Case.
b. Product Characteristics—Characteristics relating to the deliverable software product should be described using appropriate rationale. However, it is important that the information which is provided shows a clear validation of supportability achievement, rather than a mere assembly of facts or assertions. Typical aspects that might be covered are as follows:

1. Sizing data (of source code, object code);
2. Complexity/modularity or other quality metrics;
3. Functional partitioning;
4. Language features;
5. Use of common/neutral interface standards;
6. Delivery media, and files and data formats;
7. Installation and configuration; and,
8. User manuals and design documentation standards (if deliverable).

c. Process Characteristics—Supportability characteristics relating to the engineering and management processes used in development, delivery, and/or operation of the software product should be described. Where such factors relate to standards or practices which are applied universally by the supplier, an external reference or summary/extract from the appropriate corporate documentation may be adequate. However, if this approach is taken, it should nonetheless be made clear how such processes have materially affected the supportability of the subject software product. If a supplier's organization and development processes have received independent assessment against a non-proprietary formal standard, information may be included to identify the reference standard and the scope of any associated certification which has been awarded.

d. Product Support Infrastructure—Information may be provided to show that the product support infrastructure proposed or established by the supplier will meet the customer's requirements for quality of service at an acceptable cost. This may embrace both direct support for maintenance and upgrade of delivered software, and services to provide help to users on system operation or training. For developed software, information may also be required concerning how the quality of support will be assured during the transition phase from development to full in-service operation.

6.3.1.3 *Types of Evidence*—A Software Supportability Case may be based on a variety of types of evidence. The method used in a particular instance may be chosen at the supplier's discretion, as appropriate to the nature of each requirement which is addressed. Suitable approaches are described as follows. These may be used in isolation, but more typically they will be used in combination to provide an adequately robust Case.

a. Quantitative Evidence—Quantitative evidence would use defined methods of analysis to generate metrics that demonstrate the required (or desirable) supportability features in the target software product. This type of evidence would also include the results of any testing or demonstrations conducted as part of a supportability program.

b. Qualitative Evidence—Qualitative evidence would focus on the processes used for development and support of the software. It would seek to assure satisfaction of supportability requirements by inference, based on the demonstrable quality, maturity, and integrity of the underlying engineering and management processes.

c. Historical or Comparative Evidence—Historical evidence would be relevant for software already in use and supported for other customers. Comparative evidence would be relevant for software which is a variant of an existing product, or is similar to an existing product produced by the same supplier. The information provided might include both quantitative and qualitative aspects of the product and the associated support services.

6.3.2 SUPPORTABILITY RISKS/CONSTRAINTS—Where any software supportability risks are identified in the course of a supportability program, these may be described within the Software Supportability Case and linked to the relevant product/process/infrastructure characteristics. Associated information on risk assessments and risk reduction actions may also be given. Examples of issues that might be covered are:

a. Obsolescence (e.g., of software languages and tools);
b. Dependencies on unknown third-party suppliers;
c. Lack of escrow arrangements; and
d. Immaturity or uncertainty in software support cost modeling.

6.3.3 MAINTENANCE OF THE SUPPORTABILITY CASE—In general, the rationale for generating a Software Supportability Case during development will apply similarly to maintenance of the case during the in-service phase. The Case will provide the basis for assurance that the original supportability requirements continue to be met in the face of on-going evolution and change to the software. However, maintenance of the case will incur additional cost. Whether such costs are justifiable, and how maintenance should be managed, is a matter for individual projects. Where a Case relates to a custom software development, such decisions should be taken jointly between the supplier and the customer.