JSP 886
THE DEFENCE LOGISTICS SUPPORT CHAIN MANUAL

VOLUME 7
INTEGRATED LOGISTICS SUPPORT

PART 4
SOFTWARE SUPPORT

THE MASTER VERSION OF JSP 886 IS PUBLISHED ON THE DEFENCE INTRANET.

FOR TECHNICAL REASONS, EXTERNAL LINKS ON THIS INTERNET VERSION HAVE BEEN REMOVED.

VERSION RECORD		
Version Number	**Version Date**	**Description**
1.0	21 Jul 06	Initially published as part of JSP 586 Volume
1.1	28 Aug 08	Revised to reflect DE&S organizational changes
2.0	29 Jan 10	Revised to include additional advice & guidance

CONTENTS

ANNEXES

INTERNET VERSION – MASTER IS ON THE DEFENCE INTRANET

SOFTWARE SUPPORT

CONTEXT

1. This Part contains subject matter provided by the Defence Equipment and Support (DE&S), Systems Engineering and Integration Group (SEIG), Software Supportability (SS) Team. The purpose of this Part is to define authoritative policy and provide guidance to maximise the likelihood of procuring and maintaining supportable software. It supersedes software policy and information previously contained within:

   a. Air Publication (AP) 100D-10.

   b. Joint Air Publication (JAP) 100A-01 Chapter 12.8.

2. Software may provide a wide variety of functions at various levels within a system's physical structure. As such, all software will require appropriate through-life support in order to sustain operational effectiveness of the host system. This publication is applicable to software employed in the Land, Sea, Air and Information System domains and should be considered across all Defence Lines of Development (DLODs).

POLICY

3. It is MOD policy that the support of software shall be afforded full consideration throughout the equipment lifecycle. This shall be enabled by:

   a. The appointment of an authoritative and competent software supportability Subject Matter Expert (SME).

   b. The application of SSA as detailed within JSP 886 Volume 7 Part 3.

      [Note] The depth to which analysis is carried out shall be variable according to the phase of the equipment lifecycle. Justification shall be provided for every SSA reduced effort activity.

   c. The establishment of traceable and measurable software support requirements, as defined via SSA activities, which are included in formal requirements documentation.

   d. Progressive assurance to demonstrate the achievement of software support.

4. The Project Team Leader (PTL) is responsible for:

   a. The implementation and maintenance of Software Support Policy for the project i.a.w. the Integrated Logistic Support Manual.

   b. Ensuring that changes to software are progressed in a controlled manner.

PRECEDENCE AND AUTHORITY

5. The authority to carry out Software Support as an element of the ILS process and methodology is promulgated from Standing Instruction 10 Support Solution Development.

MANDATED REQUIREMENTS

6. There are no mandated requirements associated with software management. However, failure to address software support issues will impact on lifecycle costs, performance, availability, maintainability and possible overall system safety.

ASSURANCE

7. Software Support is an element of the ILS process that is independently assured against Governing Policy 2.1 and Governing Policy 2.5. Guidance for Assurance can be found in JSP 886 Volume 1 Part 3 Support Solutions Envelope.

8. The Support Improvement Team, independently identifying risks to delivery and assisting in the provision of a coherent support solution, externally assesses Governing Policies.

PROCESS

9. Procedures and guidance for Software Support are outlined in JSP 886 Volume 7 Part 3 Logistic Support Analysis Guide.

KEY PRINCIPLES

The Role Of Software

10. The idea that software can be developed and finished, without further modification, is false. Systems are designed to complete functions within a constantly changing world and they are subject to a constant flow of possible change drivers. A significant proportion of functionality in modern systems is enabled by the use of software. As such, the use of software is appropriate as:

a. It can perform highly complex tasks without adverse impact on system size and weight constraints.

b. The modification of software can be conducted with minimal disruption to operational availability.

11. In common with all other design disciplines, and taking a whole-life perspective, the support of software should be understood, managed, and afforded a level of consideration commensurate with its role and the financial commitment it attracts. The reasons for this are:

a. Software support is an essential component of the Defence Strategic Vision for the provision of sustainable capability at optimum Whole Life Costs (WLC)/Cost of Ownership (COO).

b. System reliability will, in part, be reliant upon software reliability.

c. The continuing evolution in military tactics and computing technology demands highly flexible and responsive support arrangements.

Software Support

12. The term ‘software support’ describes those activities that enable and sustain system software. Software Support consists of 2 distinct elements:

a. **Software Operations Support (SOS)**. Refers to those actions necessary to Load, Re-load, Download, Replicate, Store, Distribute or carry out any software handling activity.

b. **Software Modification**. Refers to the development and implementation of a design change to an In-Service software item. Some form of 'Request for Change' should always initiate software modification.

[Note] All software change is termed modification because the act of changing software results in the creation of a new software product. Specifically, it is not returned to its original state.

13. These 2 elements, although related, are distinct in regard to support processes, resources and facilities. The application of Software Support Analysis (SSA), as detailed within JSP 886 Volume 7 Part 3, will assist in the identification of support requirements.

Software Support Considerations

14. Software support considerations are detailed at Annex A.

GLOSSARY/ABBREVIATIONS

15. A glossary of terms and list of abbreviations are detailed at Annexes B and C respectively.

ASSOCIATED STANDARDS AND GUIDANCE

16. The following standards and guidance are provided for this Part and, unless otherwise stated, are at the latest amendment state:

a. BS ISO/IEC 12207 - Information Technology, Software Life Cycle Processes.

b. BS EN 61508-4:2002 - Functional Safety of Electrical / Electronic / Programmable Electronic Safety-Related Systems Part 4 - Definitions and Abbreviations.

c. ISO 9001:2000 - Quality Management System. Requirements.

d. TickIT Guide - A Guide to Software Quality Management System Construction and Certification to ISO 9001:2000.

e. Defence Standard 00-49 - Reliability and Maintainability MOD Guide to Terminology Definitions.

f. Defence Standard 00-56 - Safety Management Requirements for Defence Systems.

g. Defence Standard 05-57 - Configuration Management of Defence Material.

h. JSP 440 - The Defence Manual of Security Part 8 - Communications Security.

i. JSP 886 Vol 7 Pt 2 - Integrated Logistic Support Manual.

j. Capability Maturity Model Integration (CMMI), Software Engineering Institute (SEI).

k. Military Air Environment (MAE) Business Procedure (BP) 1201 - Guide To The Acquisition Safety And Environmental Management System.

OWNERSHIP

17. This policy is sponsored by the SEIG Head of Software Systems and provides advice and guidance to the MOD Community on software support issues associated with any stage of the equipment lifecycle.

Contact details
Hd of Software Support, SEIG, DGS&E
E-mail:
Internal: DES SE SEIG-SSTL
External: billy.coffield914@mod.uk
Tel Civ: +44(0) 7795 044322
Tel Mil: 9355 72379

Document Editor
DES JSC TLS-POL-PC5
E-mail:
Internal: DES JSC TLS-POL-PC5
External: desjsctls-pol-pc-5@mod.uk
Tel Civ: +44(0) 1225 882891
Tel Mil: 9355 82891

SOFTWARE SUPPORT CONSIDERATIONS

THE NATURE OF SOFTWARE FAILURE

1. In all but the most elementary software there will be faults present to some extent throughout its life; however, unlike hardware[1], software only fails systematically[2]. As such, failures attributable to software are caused by:

   a. Faults that have not been removed prior to delivery.

   b. Operation of the system outside of its specified limits.

   c. The introduction of faults through subsequent maintenance.

2. To remove a software fault, modification is required. However, to retain a system in, or return it to a previously specified condition, recovery activities (reload / reboot / restart) are necessary.

SUPPORT CONSIDERATIONS

3. The inability to support software can prevent software evolution, which can ultimately expose all stakeholders to the risk of a capability gap. To reduce such risks, achieving supportable software should be seen as a design goal, and the adoption of a structured approach to design should ensure that:

   a. A view across all DLODs is taken to understand the interrelationships of software within the context of other elements.

   b. Stakeholder analysis is carried out which identifies factors affecting the development, purchase, and operation of software.

   c. A comprehensive review is undertaken for initial identification of through-life software change needs and associated software support requirements.

   d. A whole-life approach is adopted, which considers the planning and costing of through-life software change and associated In-Service[3] software support requirements.

   e. Support processes are reviewed and updated to reflect the maintenance and modification requirements of the evolving system.

4. **Support Planning**. During acquisition, all planning and costing associated with software support should be documented as part of through-life management planning and included as part of the Whole Life Cost (WLC) strategy. Once In-Service, any changes to software support must be reflected in the appropriate documentation.

---

[1] In hardware, random failures are dominant and drive maintenance activities.
[2] Failure related in a deterministic way to a certain cause, which can only be eliminated by a modification of the design or of the manufacturing process, operational procedures, documentation or other relevant factors.
[3] The support of software associated with the In-Service phase of the CADMID/T lifecycles.

5. **Formal Agreements**. A formal agreement[4] is to be raised to define the relationship between stakeholders and their responsibilities for the provision of a software support capability. Whilst this agreement will remain extant throughout the service life of the system, it should be reviewed on a periodic and event driven basis for continuing applicability.

SOFTWARE SUPPORT ANALYSIS

6. Software Support Analysis (SSA) is a series of analytical tasks that identifies support requirements, issues and drivers as early as possible in the equipment lifecycle. The application of SSA will:

a. Enable logistic support considerations to influence the design of the equipment.

b. Define logistic support processes and their resource requirements for the life of the equipment.

7. The generic software support model detailed within Section 5 can aid the application of SSA and the identification of software support requirements. In particular, SSA should fully consider the following:

a. Types of Software Change.

b. Scaleable Support.

c. Alternative Support Approaches.

TYPES OF SOFTWARE CHANGE

8. Software changes can be classified as follows:

a. **Corrective**. A corrective change modifies a software item to remove a software fault.

b. **Adaptive**. An adaptive change modifies a software item to enable it to continue to meet its specification in a changed environment.

c. **Perfective**. A perfective change modifies a software item to enable it to meet its existing specification in an improved fashion.

d. **Enhancement**. An enhancement change modifies a software item to add additional functionality to the system.

SCALEABLE SUPPORT

9. Throughout the life of a system, software support drivers will not be of equal criticality. As such, the software support solution must be scaleable and flexible in response to the criticality of change; typically this can be facilitated by:

a. The through-life retention and timely availability of an adequate support solution, which includes the Software Development Environment (SDE) and sufficiently skilled personnel such that the capability is sustainable through life[5].

[4] e.g. Contract, Customer Supplier Agreement, Service Level Agreement, Internal Business Agreement, etc.
[5] Where software modification is allocated to multiple organisations, the duplication of support resources (SDE and personnel) must be carefully managed to ensure support efficiency is not compromised.

b. A support solution that explicitly recognises and accommodates the capacity to undertake both routine and urgent software modification.

ALTERNATIVE SUPPORT APPROACHES

10. The transformation of logistic support has resulted in a shift from Traditional based support to Contracting for Availability (CfA) and ultimately Contracting for Capability (CfC). Whilst this shift is equally applicable to software dependent systems, the implications of each approach need to be considered if support solutions are to prove effective and efficient, irrespective of the support provider. Regardless of whether support is provided by an Industry, Service manned, or partnered team; the main approaches are:

a. **Traditional**. For software, Traditional support and Spares Inclusive can be treated as equivalent. Both approaches typically rely upon the provision of a software warranty for the correction of software faults, with additional support arrangements being established for software adaptations and enhancements.

(1) **Corrective Changes**. Corrective software changes are typically facilitated under Contractor Logistic Support (CLS) arrangements. If corrective changes are carried out at the Suppliers expense, this effectively provides a software warranty. During the establishment of CLS contracts the following alternatives need to be considered:

(a) **Lifetime Warranty**. With this option, the Supplier implements corrective changes, free of charge, for the life of the software.

(b) **Limited Life Warranty**. With this option, the Supplier implements corrective changes, free of charge, for an agreed period of time after the In-Service date[6]. Once the warranty period has expired the Customer pays for all corrective changes.

(c) **No Warranty**. With this option, the Customer pays for all corrective changes and is considered undesirable.

(2) **Adaptations and Enhancements**. Software adaptations and enhancements are often required to sustain capability. However, unlike corrective changes, financial responsibility for their implementation will almost always belong to the Customer. A Post Design Services (PDS) arrangement is often established to facilitate the through-life adaptation and enhancement of software products.

b. **Contracting for Availability**. With CfA, the Supplier holds responsibility for the timely provision of a serviceable system against a predefined specification. As such, only software changes that sustain system availability should be carried out under CfA arrangements. Where the need for MOD driven software adaptations and enhancements exists, a separate PDS arrangement is still required.

(1) **Corrective Changes**. The application of CfA to software will only be effective if the measure of system availability includes all failures that are attributable to software. As such, the success of CfA is highly dependent upon the establishment of appropriate and robust measures of software availability. Successful CfA will:

(a) Negate the need for a software warranty and CLS.

[6] To be effective the warranty period must expose the greatest possible number of software faults. This will be assisted by diverse system usage, likely of exercising the full range of functions and operational scenarios.

(b) Incentivise the Supplier to establish and maintain an effective and efficient support solution.

(2) **Adaptations and Enhancements**. As adaptations and enhancements alter system specification, it can be inappropriate[7] for these types of changes to be carried out under CfA arrangements. Furthermore, where a system's specification is changed, CfA arrangements might need to be reviewed and possibly updated.

c. **Contracting for Capability**. In addition to the support provided by CfA, a CfC support solution would be responsible for the timely delivery of a capability as opposed to a predefined system. As such, CfC must accommodate all types of software change throughout a system's life. To facilitate CfC, both the Customer and Supplier must establish a good understanding of the potential need for software change. Where the need for change is misunderstood, the through-life provision of capability will be jeopardised.

11. The contracting mechanism for software CfA and CfC will be different than that of Traditional support. For CfA, the contracting processes must be able to accommodate changes to availability criteria in response to system specification changes. For CfC, the contracting processes must be able to accommodate changes to system specification in response to evolving operational needs.

SUPPORT SOLUTION

12. Based on the outputs of SSA, the chosen software support solution must address, as a minimum, the following:

a. The Sponsor and User requirements.

b. Areas where it is imperative that MOD personnel are employed (i.e. to maintain the "Intelligent Customer" position).

c. The justification of either MOD or Industry support, or a mixture of both (partnering).

d. A full definition of the level of support to be provided (for both routine and urgent operations).

e. The appropriate software support approach (i.e. CLS, CfA, etc.).

f. The software support infrastructure required (i.e. personnel, facilities, training, etc.).

g. The provision of support at optimal cost.

h. Risk management.

i. The support requirements for any associated data that enables operational use of system software[8].

j. Any other project specific factors relating to, or impacting upon, software support.

[7] MOD driven changes should be facilitated under separate PDS arrangements. However, supplier driven changes to meet system availability criteria should be incorporated within CfA agreements.
[8] Where data support is required, it will typically be considered as an integrated function with the overall software support requirements.

SOFTWARE SUPPORT REQUIREMENTS

13. Irrespective of software type, ranging from Commercial Off The Shelf (COTS) to bespoke development, software support will always comprise the same generic functions. However, the organisation that carries out each function will vary greatly depending upon the type of software and need for support.

14. The software support model detailed at Figure 1 illustrates the generic support functions, and the flow of items that enable software support. As a generic model, it can be utilised to assist with the elicitation of software support requirements specific to an individual project's needs. By defining complete, appropriate, and measurable support requirements, the likelihood of establishing effective and efficient software support solutions is improved, and the ability to sustain capability maximised.

15. Once the need for software support is identified, high-level support requirements must be placed in the User Requirements Document (URD), such that detailed support system requirements can be placed in the System Requirements Document (SRD). SRD software support requirements must define the required level of service that the support solution is to provide. This can be measured in terms of:

a. End-to-end software modification timescales (for both routine and urgent changes).

b. The response times and maximum acceptable durations for each support function.

c. The maximum acceptable duration for items to pass from one support function to another.

d. Support Function usage rate (i.e. the number of times each function is used per year).

16. Descriptions of the actors illustrated at Figure 1 are as follows:

a. **Users**. The term 'Users' refers to all personnel that interact with the system, specifically this includes the operators and support personnel. As such, Users may generate questions about system operation, discover problems, and generate ideas for adaptations and enhancements, which are collectively referred to as queries. These queries are formalized by creating a Query Report, which represents all internally generated change needs. The Query Report documents all relevant information and data, and is forwarded to the Query Evaluation function.

b. **External Change Drivers**. The term 'External Change Drivers' refers to a source of change needs that does not originate from User operation. Examples of these change drivers may be the need to preserve functionality in response to changes in interfacing software components, or underlying hardware.

c. **Host System**. The term 'Host System' refers to the physical equipment on which the software and data resides, such that through its operation, system functionality or capability is provided.

SOFTWARE SUPPORT MODEL

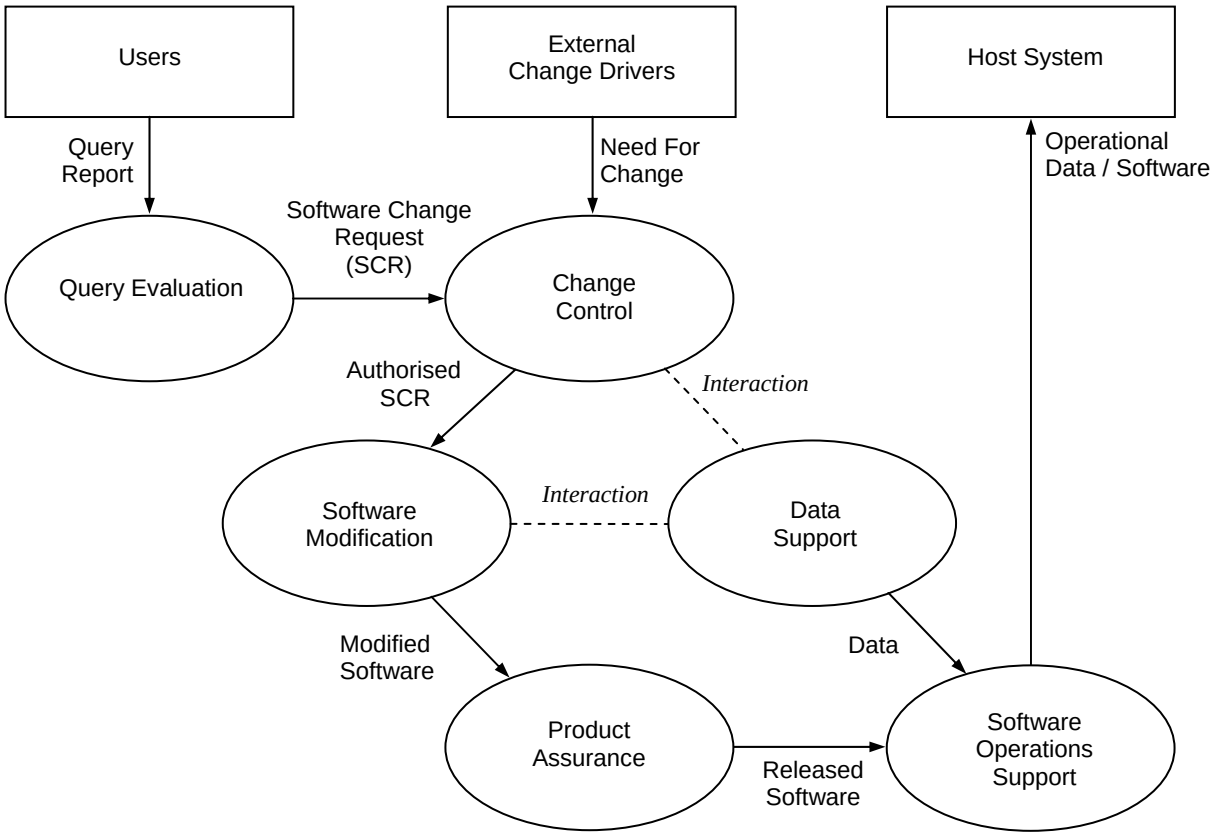

Figure 1 - Software Support Model

17. Descriptions of the functions illustrated at Figure 1 are as follows:

a. **Query Evaluation**. The Query Evaluation function evaluates and filters Query Reports to identify the cause of any query, categorize the nature of any problems, remove duplication and identify User error. The function can solve many problems without the need for software modification by identifying workarounds; however, some queries will generate a Software Change Request (SCR). For all SCRs, the evaluators must assess the operational benefits, costs and risks of each change and report its findings to the Change Control function and the originating organisation. The Query Evaluation provides feedback to the Users as to the progress of all Query Reports, and Change Control as to the outcome of feasibility and impact analysis.

b. **Change Control**. The Change Control function authorises and prioritises SCRs to meet the needs of operational capability and system readiness. Change Control deals with user-initiated SCRs (via evaluated Query Reports) as well as externally driven software change needs. The function reconciles demands for change within the goals, constraints and available resources imposed upon it. Where necessary, Change Control will task Query Evaluation to carry out feasibility and impact studies on externally driven software changes. Due to the dependencies between software and data, the Change Control function must be aware of all software and data changes, and their potential impact upon system functionality.

c. **Software Modification**. The Software Modification function is responsible for the implementation of authorised SCRs. The output of Software Modification is a new software

load and supporting documentation, which after its release is ready for use by the Host System. Due to the dependencies between software and data, the Data Support function has to be advised of all software changes.

d. **Data Support**. Within the context of the model (Figure 1), Data refers to information, both mission and engineering related, that is loaded or downloaded from the Host System, or used to configure the Released Software. The Data Support function is responsible for the creation, preservation, analysis and modification of data. The output of Data Support is a new data load and supporting documentation for use within the Host System. Due to the dependencies between data and software, the Software Modification function has to be advised of all data changes.

e. **Product Assurance**. The Product Assurance function is responsible for verifying that modified products are acceptable for release, i.e. they remain acceptably safe, secure, reliable or supportable for use after the implementation of authorised SCRs. At its basic level it will ensure Released Software is configured correctly and appropriately documented on release paperwork. It is important to realise that this function only represents the formulation of evidence into a statement of assured integrity for a desired quality characteristic. Specifically, the function does not represent all the activities that build towards product integrity, as these activities exist throughout the maintenance model.

f. **Software Operations Support.** The definition of Software Operations Support (SOS), detailed at Paragraph 8a and reproduced below;

> *"SOS refers to those actions necessary to Load, Re-load, Download, Replicate, Copy, Store, Distribute and carry out any software handling activity."*

can be expanded to include all the actions necessary to load, configure, retrieve and sanitize mission or engineering data. This expansion is appropriate where systems are dependent upon the manipulation of data for any operational or engineering output.

SOFTWARE CHANGE MANAGEMENT

18. Software within any system will be subject to change throughout its life. Such software changes could be initiated by software faults, changes to other systems with which the software interacts, or the need for capability enhancement. In order to maintain system integrity through life, it is essential that software support activities are comprehensively managed and that change is carried out in a controlled manner. Software Change Management should ensure, as a minimum, that:

a. The impact of any software change is fully assessed[9].

b. The effect of hardware modifications are fully assessed for their impact on system software.

c. The release of the software is managed, including the arrangements for system testing and acceptance.

d. Software Configuration Management (CM) and its associated data is maintained.

19. The following paragraphs outline how software change control and configuration management is to be implemented for software systems; however, it does not provide the detailed

[9] Including: Security, Safety, Training and Quality Assurance impact analysis.

procedures required[10]. The following guidance is applicable to any organisations or bodies that are responsible for software change management, such as:

a. Software Change Control Board (SCCB).

b. Agencies responsible for the support of software and data, for example:

(1) Design Authorities (DAs)[11].

(2) Engineering Authorities (EAs).

(3) Support Authorities (SAs).

(4) MOD Role Offices (ROs).

(5) Integrated Logistic Support Managers (ILSMs).

(6) Software Support Teams (SSTs)[12].

(7) Original Equipment Manufacturer (OEM).

SOFTWARE CHANGE CONTROL

20. Software change control consists of 2 distinct elements:

a. The day-to-day activities carried out by the SCCB.

b. The analysis and progression of software changes.

SOFTWARE CHANGE CONTROL BOARD

21. For every system that contains software, the establishment of a distinct SCCB should be considered[13] to provide the management structure for exercising control over changes to software contained within a system. The extent to which software will be managed by the SCCB will be dependent on the need for change. It is the responsibility of the ILSM to:

a. Establish a SCCB.

b. Generate outline Terms of Reference (TORs) for the SCCB.

[Note] Generic SCCB TORs are detailed at Appendix 1.

c. Document the SCCB membership within appropriate support policy documentation.

22. Each SCCB, as a minimum, is to have representatives from the appropriate operations specialists, the Project Team, SA, EA, the DA and/or the primary system contractor and the SST. This ensures that the SCCB has among its members the operators view and the controlling, tasking and financial authority. The SCCB discharges its responsibilities through the authority of its individual members. Although not universally applicable, typical key functions of the agencies represented on the SCCB are as follows:

[10] It is the responsibility of the SCCB to arrange for such procedures to be prepared and issued.
[11] May be Service or Industry.
[12] May be Service or Industry or a combination of each under a partnered arrangement.
[13] The system change control board may subsume the responsibility where software is perceived to have a negligible requirement for change.

a. Providing the operators view.

b. Providing the links between MOD, contractors and other system CM agencies.

c. Providing specialist advice on the feasibility of proposed software changes with consideration of cost, time-scales, release to service, security implications, safety implications and any other project specific issues.

d. Providing tasking and functional control of the programme of work.

e. Controlling the configuration of support facilities ensuring that they are maintained in-line with any upgrades to the operational system.

f. Where MOD personnel are employed within SSTs, providing an appropriate training, travel and subsistence, and accommodation budget.

g. Holding responsibility for the integrity of the system design through the fulfilment of release to service, security and safety requirements.

h. Ensuring that the master design record and associated documentation is maintained.

i. Authorising the release of software for operational use in consultation with the DA as appropriate.

PROGRESSING SOFTWARE CHANGES

23. **Software Change Request**. All proposed changes to the system software are to be strictly controlled. Accordingly, requests for changes are to be initiated by a Software Change Request (SCR), which is then forwarded to the SCCB for consideration.

24. **SCR Assessment**. On receipt of a SCR, the SCCB will generally request a feasibility study to be carried out that covers the following:

a. An assessment of whether the SCR is either operationally beneficial, or offers a potential reduction in support costs; if neither of these criteria are satisfied, the SCR should normally be rejected and the originator informed of the rejection. In all cases, the SCCB is to assess whether the SCR is affordable and represents value for money. The SCCB should allocate a priority to each SCR that is accepted.

b. Liaison with appropriate technical bodies in order to:

(1) Confirm that a software, rather than a hardware, solution is appropriate.

(2) Assess whether associated hardware modifications will be required.

(3) Assess whether planned hardware modifications will impact on the software solution to the SCR.

c. An assessment of the likely impact of an SCR upon the security accreditation or the safety case of the system. If the SCR is progressed, the SCCB must initiate action to re-accredit the security of the system or revalidate the safety case.

25. **SCR Status Elevation**. If a SCR represents a major functional change to the system requirements, then the nature of the software change will be software development, rather than modification. When significant software development is required, the SCCB should consider elevation to the appropriate Director of Equipment Capability (DEC) for higher-level direction.

CONFIGURATION MANAGEMENT

26. The effectiveness of software support depends on the ability to manage software configuration, and as such, Configuration Management (CM) must be a continual and omnipresent process throughout the life of the system. CM should be applied to system software as a whole utilizing the following 5 basic activities:

a. **Planning**. Plan and define the purpose, scope, objectives, polices and procedures of the CM process and any associated CM Database (CMDB).

b. **Identification**. The selection and identification of all identified Configuration Items (CIs) within the system.

c. **Control**. Assurance that only authorised and identifiable CIs are accepted and recorded from receipt to disposal.

d. **Status Accounting**. The reporting of all current and historical data concerned with each CI throughout its life cycle.

e. **Verification and Audit**. A series of reviews and audits that verifies the physical existence of CIs, and checks that they are correctly recorded in the CMDB.

27. Additionally, within the modification process, all SCRs are to be brought under configuration control and remain under configuration control even if they are subsequently rejected[14].

FURTHER SUPPORT CONSIDERATIONS

SAFETY ISSUES[15]

28. As part of any software change, the impact on the system safety case must be considered and managed.

SECURITY ISSUES[16]

29. As part of any software change, the impact on the security accreditation set must be considered and managed.

TRAINING ISSUES

30. The requirements of any core competencies relating to the support and operation of software must be identified within the software support management strategy. The ILSM will identify the specific project requirements and standards applicable to the training of personnel during the early application of Software Support Analysis, thereby ensuring that appropriate training is provided in a timely manner.

[14] So that duplication of effort can be minimised over time.
[15] The specific requirements regarding Safety are detailed within Defence Standard 00-56.
[16] The specific requirements regarding Security are detailed within JSP 440 Part 8.

QUALITY ASSURANCE[17]

31. It is essential that software modification activities do not degrade the integrity of the system. To this end, it is incumbent upon the IPT and SA to identify the requirements of Software Quality Assurance (SQA), and any third party certification to be applied. Where the MOD is partnered with Industry, the application of SQA standards should be consistent wherever practicable to:

a. Provide a common framework of SQA concepts and instructions.

b. Facilitate the transfer of software and data between the MOD and Industry with minimum rework.

c. Satisfy the SQA related requirements for gaining software release to service.

[17] For further advice on Logistic Quality Policy contact the Defence Quality Agency (DQA).

OUTLINE TERMS OF REFERENCE FOR A SOFTWARE CHANGE CONTROL BOARD

1. Outline Terms of Reference (TOR) for a Software Change Control Board (SCCB) are as follows:

a. To draft, agree and periodically review SCCB procedures.

b. To manage software changes with cognisance of higher-level system configuration control board requirements.

c. To consider the effects of software changes on safety, security, system compatibility, and operational role (including interoperability with other systems).

d. To allocate priorities to Software Change Requests (SCRs).

e. To consider the requirement for system validation and operational trials.

f. To assist in the release of all new software.

g. To advise higher authorities of new software releases and ensure a formal record of software configuration is maintained.

2. To fulfil its obligations, the SCCB may decide that subordinate committees are required to provide specialist advice on certain issues. For example, a sub-committee may be formed in order to assess all SCRs for engineering specialisations such as Safety and Security. This sub-committee could be granted authority to task a SST to conduct feasibility studies into the cost or time implications of a SCR. In such a case, the SCCB is to define the membership of the sub-committee and provide them with detailed TOR and procedures. However, responsibility for the authorisation of rejection of SCRs always rests with the SCCB.

3. Overarching change management for an operational system typically rests with a system Change Control Board (CCB). Where such a system CCB exists, the SCCB is subordinate to that committee and will give specialist advice on software modifications. Where a system configuration control board does not exist, the ILSM responsible for establishing the SCCB is to define its relationship with other associated configuration bodies.

Annex B to
JSP 886, VOL 7, PART 4
Dated 29 Jan 10

GLOSSARY OF TERMS

1. The definitions applicable to the context of this Part are:

a. **Data**. Representation of information both mission and engineering related, loaded to or from the Host System. Data may take the form of static information, such as geographical information, or it may take the form of instructions to specify mission objectives.

b. **Host System.** The physical equipment in which the software and data resides, such that through its operation some function of the system is enabled.

c. **In-Service Software Support**. The support of software associated with the In-Service phase of the CADMID/T lifecycles.

d. **Random Hardware Failure**[18]. Failure, occurring at a random time, which results from one or more of the possible degradation mechanisms in the hardware.

e. **Software.** Programs, procedures, rules, data and any associated documentation pertaining to the operation of a computer system.

f. **Software Lifecycle**. The activities necessary from initial inception to develop, maintain and finally dispose of software and its associated data.

g. **Software Operations Support (SOS)**. The actions necessary to Load, Re-load, Download, Replicate, Copy, Store, Distribute or carry out any software handling activity.

h. **Software Modification**. Software Modification is the development and implementation of a design change to an In-Service software item. A Software Change Request should always initiate software modification action.

i. **Software Support Team (SST)**. An organisation established to provide the In-Service software support and may be wholly MOD provided, wholly Industry provided, or a partnered arrangement between the MOD and Industry.

j. **Stakeholders**. Agencies and Users who have influence over or are influenced by the software support decisions.

k. **Systematic Failure**[Error! Bookmark not defined.]. Failure related in a deterministic way to a certain cause, which can only be eliminated by a modification of the design or of the manufacturing process, operational procedures, documentation or other relevant factors.

l. **Users**. An individual or organization that uses the host system to perform a specific function.

[18] BS EN 61508-4:2002 - Functional safety of electrical / electronic / programmable electronic safety-related systems Part 4 - Definitions and Abbreviations.

ABBREVIATIONS

1. The abbreviations applicable to the context of this Part are:

AP	Air Publication
BP	Business Procedure
CfA	Contracting for Availability
CfC	Contracting for Capability
CCB	Change Control Board
CI	Configuration Item
CLS	Contractor Logistic Support
CM	Configuration Management
CMDB	Configuration Management Database
CMMI	Capability Maturity Model Integration
COO	Cost of Ownership
COTS	Commercial Off The Shelf
DA	Design Authority
DE&S	Defence Equipment and Support
DEC	Director of Equipment Capability
DLODs	Defence Lines of Development
EA	Engineering Authority
ILSM	Integrated Logistic Support Manager
JAP	Joint Air Publication
MAE	Military Air Environment
OEM	Original Equipment Manufacturer
PDS	Post Design Services
PTL	Project Team Leader
RO	Role Office
SA	Support Authority
SCCB	Software Configuration Control Board
SCR	Software Change Request
SDE	Software Development Environment
SEI	Software Engineering Institute
SEIG	Systems Engineering and Integration Group
SME	Subject Matter Expert
SOS	Software Operations Support
SQA	Software Quality Assurance
SRD	System Requirements Document
SS	Software Supportability
SSA	Software Support Analysis
SSE	Support Solutions Envelope
SST	Software Support Team
TORs	Terms of Reference
URD	User Requirements Document
WLC	Whole Life Costs