

**WESTLAND HELICOPTERS LIMITED
YEOVIL SOMERSET**

**REQUIREMENTS for HIGH INTEGRITY (non-airborne),
VALIDATION, TEST and SIMULATION SOFTWARE**

Document No: SDDR-301
Issue: 2
Date: MAR 1992

Prepared by: J LONGLEY
Chief Quality Engineer
(Software)

Approved by: K G BANNISTER
Chief Systems Designer

Approved by: A H VINCENT
Head of Engineering

Approved by: P A JENNINGS
Supplier Quality Assurance
Manager

Copyright Westland Helicopters Limited 1992

Copyright of this document is vested in Westland Helicopters Limited (WHL), and it may not be copied or used for any other purpose than for which it is supplied, or disclosed to any third party without the express authority of WHL.

AMENDMENT SHEET

ISSUE NUMBER	DATE OF ISSUE	REASON FOR AMENDMENT	RAISED BY
1 (DRAFT)	AUG 1990	Initial Issue	J Longley
1 (DRAFT)	NOV 1990	2nd DRAFT	J Longley
1	NOV 1990	Formal Issue	J Longley
2 (DRAFT)	JAN 1992	Annual Review including full implementation of ISO 9001 & ISO 9000-3	J Longley
2	MAR 1992	Formal Issue	J Longley

THIS PAGE LEFT INTENTIONALLY BLANK

SDDR-301 Issue 2
CONTENTS

1	INTRODUCTION	1
1.1	Purpose	1
1.2	Scope	1
1.2.1	Applicability	1
1.2.2	Superseded Standards	2
1.3	Applicable Standards	2
1.4	Other Related Standards	2
1.5	Associated Policies and Documents	2
1.6	Interpretation	3
1.7	Control and Use of Document	3
1.8	Overview Of Contents	3
2	GENERAL REQUIREMENTS	5
2.1	Roles and Responsibilities	5
2.1.1	Authorities	5
2.1.2	Customer	5
2.1.3	Contractor	5
2.1.4	Contract	5
2.1.5	Project	5
2.1.6	Task	5
2.2	Software Classification: Usage and Risk	5
2.3	Acceptance and Approval	6
2.3.1	General Requirements	6
2.3.2	Documentation	6
2.3.3	Equipment and Software	6
2.4	Project and Software Development Environment	7
2.4.1	General Requirements	7
2.4.2	Design Methodologies	7
2.4.3	Proprietary Design Methodologies	7

2.4.4	Performance and Adequacy	7
2.5	Proprietary Software Management	8
3	PROJECT PHASES AND SOFTWARE LIFE CYCLE	9
3.1	Project Phases	9
3.2	Software Development Life Cycle	9
3.3	Development Support Activities	9
	FIGURE 1 DOD-STD-2167A System Development Life Cycle.	10
4	SOFTWARE DEVELOPMENT ACTIVITIES	11
4.1	General Requirements	11
4.2	System Requirements Analysis/Design	11
4.3	Software Requirements Analysis	11
4.4	Preliminary Design	11
4.5	Detailed Design	12
4.6	Coding and Computer Software Unit (CSU) Testing	12
4.7	Computer Software Component (CSC) Integration and Testing	12
4.8	Computer Software Configuration Item (CSCI) Testing	12
4.9	System Integration and Testing	13
	FIGURE 2 Software Development Activities and Associated Products	14
5	SOFTWARE DEVELOPMENT MANAGEMENT ACTIVITIES	15
5.1	Planning	15
5.2	Reporting	15
5.3	Monitoring and Control	15
5.3.1	Meetings	16
5.3.2	Control Room/Document	16
5.4	Subcontracting	16
6	SOFTWARE QUALITY ASSURANCE ACTIVITIES	17
6.1	Planning	17
6.2	Reporting	17
6.3	Audits	17

SDDR-301 Issue 2

6.4	Reviews, Walkthroughs and Inspections	17
6.5	Fault Analysis and Corrective Actions	18
7	SOFTWARE CONFIGURATION MANAGEMENT ACTIVITIES	19
7.1	Planning	19
7.2	Item Identification	19
7.3	Status Accounting and Control	19
7.4	Issue and Release	19
7.5	Software Development Library	19
7.6	Software Archive	20
7.7	Change Control	20
7.8	Proprietary Software	20
8	SOFTWARE CLASSIFICATION AND SAFETY ACTIVITIES	21
8.1	Requirements and Planning	21
8.2	Reporting	21
8.3	Working with the Safety Authority	21
8.4	Hazard Analysis	21
8.5	Design Requirements	21
9	DOCUMENTATION	22
9.1	General Requirements	22
9.1.1	Planning	22
9.1.2	Deliverable and Non-deliverable Documentation	22
9.1.2.1	Reduction or Merger of Deliverables/Documents	23
9.1.2.2	'Electronic' Delivery	23
9.1.3	Maintenance and Updates during Development	23
9.1.4	Software Development Files	23
9.2	Management and Planning Documentation	24
9.2.1	Software Development Plan	24
9.2.2	Software Quality Assurance Plan	24
9.2.3	Documentation Plan	24

9.2.4	Software Configuration Management Plan	24
9.2.5	Software Test Plan	24
9.2.6	Acceptance Plan	24
9.2.7	Safety Plan	24
9.2.8	Verification Traceability Matrix	25
9.2.8.1	Verification Status Matrix	25
9.2.8.2	Test Traceability Matrix	25
9.3	Technical Documentation	25
9.3.1	Requirements Documentation	25
9.3.2	Design Documentation	25
9.3.3	Test Documentation	25
9.3.4	User Documentation	25
9.3.5	Issue and Release Documentation	26
9.3.6	Maintenance and Support Documentation	26
9.3.7	Software Development Documents and Procedures Manual	26
9.4	Documentation Equivalence Tables	26
10	SOFTWARE METRICS	29
10.1	General Requirements	29
10.2	Typical Management Metrics	29
10.3	Typical Size and Performance Metrics	31
10.4	Typical Reliability Metrics	31
10.5	Typical Quality Metrics	31
11	TAILORING OF THESE REQUIREMENTS FOR EXISTING EQUIPMENTS AND SOFTWARE	33
11.1	Establishing Level of Software Reuse	33
11.2	Minimum Mandatory Requirements	33
APPENDIX	A DOD-STD-2167A and Starts Life Cycle Equivalence Table	36
APPENDIX	B Glossary	37
APPENDIX	C Abbreviations	40
APPENDIX	D1 Software Quality Assurance Plan	41

SDDR-301 Issue 2

APPENDIX	D2	Documentation Plan	44
APPENDIX	D3	Software Configuration Management Plan	45
APPENDIX	D4	Acceptance Plan	47
APPENDIX	D5	Safety Plan	49
APPENDIX	E	Instrumentation Software	50
APPENDIX	F	Tables for Classification of Software	51

1 INTRODUCTION

1.1 Purpose

The purpose of this document is to identify the general requirements that shall be applied during the design, development and implementation of software generated by or for Westland Helicopters Ltd(WHL). It is to ensure that it is of the requisite quality and minimises the resources required during the software life cycle. It is intended to achieve the following:-

- a Heads of departments and their personnel understand and agree the customers requirements with respect to the development and support of high integrity software.
- b Ensures consistency of documentation from all departments involved in the production and support of such software.
- c. Ensure departments apply adequate levels of control to the development and support of such software.
- d Ensure departments provide adequate levels of visibility to the various customers or users of such software.
- e Promote the use of software engineering practice supported by suitable methods and tools.
- f Promote the use of High Order Languages, unless the contractor can demonstrate the advantages of implementation in other languages.
- g Ensure software developed for any customer is of satisfactory standard in terms of:
 - Testability
 - Integrity
 - Reliability
 - Maintainability
 - Supportability
 - Where contractually required, growth capacity and expandability shall be achieved at entry into service.

In defining the general requirements; relevant procedures, standards or code of practices shall be produced covering the design, development, implementation and support of software. Depending on the level of risk associated with a software development programme, discussed in Section 2.2, the requirements defined by this document may be tailored with agreement from the customer.

1.2 Scope

1.2.1 Applicability

This Document shall apply to **all** departments within WHL responsible for producing, procuring and/or supporting high integrity (non-airborne) software, or software utilised in the support of testing and validation of systems/subsystems and products subject to development by WHL.

This document applies to software, including firmware, as described below:-

- a All new products developed or procured by the WHL departments.
- b All existing products which require modification to meet new customer requirements.
- c Deliverable (Internal & External) and non-deliverable (test and support) software.

1.5 Associated Policies and Documents

This document draws on MOD and industry policy and recommended good practice as defined within the following documents:

- a STARTS Purchasers Handbook Edition 2
- b TickIT -
Guide to Software Quality Management System Construction and Certification Using ISO 9001

Departments should follow these policy requirements and guide-lines unless they can show 'just cause' to do otherwise in terms of:

- risk
- cost
- timescale

1.6 Interpretation

This document differentiates between mandatory requirements which are defined by the use of the term 'shall' and non-mandatory requirements which are aims or desirables which are defined by the use of the term 'should'.

Departments within WHL shall comply with the requirements of this document and applicable standards and guide-lines.

1.7 Control and Use of Document

This document is controlled by the WHL Chief Quality Engineer (Software) and is approved for issue by the WHL Manager Supplier Quality Assurance.

Any person involved with the generation of software, as covered by SDDR-301 (this document), may apply to the WHL Chief Quality Engineer (Software) proposing changes for consideration and action. Re-issues or amendments will be approved in the same manner as the original issue.

All users of this document shall ensure the latest issued version is being utilised for planning, task, or new contract work.

1.8 Overview Of Contents

- Section 1** is this introduction, which outlines the purpose and scope of the document and identifies other relevant standards.
- Section 2** defines general requirements covering overall roles and responsibilities, the classification of software, acceptance and approval of deliverable items and project and software development environments.
- Section 3** identifies software development activities and development support activities based on the DOD-STD-2167A life cycle.
- Section 4** defines the objectives of each software development activity called for by DOD-STD-2167A. The detailed requirements for these activities are referenced to the appropriate sections in DOD-STD-2167A and are not reproduced in this document.
- Section 5** defines the tasks to be conducted as part of software development management, including planning, reporting, monitoring and control, subcontracting and the use of meetings and control room/document.
- Section 6** defines the tasks to be conducted as part of software quality assurance, including planning, reporting. Audits and the use of reviews, walkthroughs and inspections.

SDDR-301 Issue 2

Section 7 defines the tasks to be conducted as part of software configuration management, including planning, item identification, status accounting and control, issue and release, the use of software libraries and archives, and change control.

Section 8 defines the tasks to be conducted as part of software criticality and safety activities, including planning, reporting, working with the Safety Authority, hazard analysis and design requirements.

Section 9 defines all documentation that must be produced as part of the development programme and identifies which shall normally be delivered. This includes management and planning documentation and technical documentation. The table within Section 9, identifies common documentation between this document, DOD-STD-2167A, and JSP188 Ed 4.

Section 10 defines a set of software metrics based upon COCOMO and of which an agreed subset shall be tracked and reported by the contractor throughout the project.

Section 11 identifies sections of this document which are deemed mandatory or tailorable when modifying/changing existing equipments and software.

Appendices

Appendix A identifies equivalence between DOD-STD-2167A and STARTS software development life cycles.

Appendix B is a glossary of terms used by this document.

Appendix C is a list of abbreviations used by this document.

Appendix D provides contents lists for a number of documents called for by Section 9.

Appendix E provides minimum requirements for Flight Approval of Instrumentation/Designvalidation software.

Appendix F provides tables for the classification of software in terms of Usage and Risk.

2 GENERAL REQUIREMENTS

2.1 Roles and Responsibilities

2.1.1 Authorities

In the context of the contract or project, the applicability, responsibility and interfaces between the following authorities shall be agreed and documented by the contractor in the Software Quality Assurance Plan, or appropriate Departmental Instruction, at the outset of the contract/project. (The Software Quality Assurance Plan is described in Section 9.2.2)

- | | | |
|---|---------------------------------------|--------------------------------------|
| a | System Design Authority or Department | (typically WHL) |
| b | System VV&T Authority or Department | (typically WHL) |
| c | Acceptance Authority or Department | (typically WHL or external customer) |
| d | Safety Authority | (typically WHL or external customer) |

The responsible Heads of the relevant Departments or their nominated representative shall be responsible for the approval of all software documentation as detailed in Section 2.3.2.

2.1.2 Customer

In the context of this document the Customer is the agency procuring goods and services either from WHL, or between or within WHL departments.

2.1.3 Contractor

In the context of this document the Contractor is the agency providing goods and services to the customer.

2.1.4 Contract

In the context of this document a contract is defined as the formal documented agreement between the Customer and the Contractor to produce major work packages in accordance with defined requirements.

2.1.5 Project

In the context of this document a Project is a planned undertaking to present results at a specified time in accordance with the controls and requirements laid down in the contract.

2.1.6 Task

In the context of this document a Task is a job of work that can be both measured in resource and linked to a planned undertaking.

2.2 Software Classification: Usage and Risk

Software shall be classified as defined within Section 8 and Appendix F. The Usage classification shall be used to determine the level of verification and test activities required as defined within this document.

Task Risk shall be measured in terms of:

- technical innovation
- technical complexity
- schedule
- project size
- importance to aircraft programme
- level of re-use of existing software

SDDR-301 Issue 2

Tables 1 and 2 of Appendix F provide criteria for the classifying of software against these factors. The level of risk shall be used to help determine the level of visibility required by the customers technical and managerial representatives. This will determine the degree of tailoring of these requirements which may be applied.

In the context of the project or task the software classification shall be agreed with the customer and documented by the contractor in the Software Development Plan at the outset of the project/task. (The Software Development Plan is described in Section 9.2.1)

In cases where software of differing classification exist within one task, the task shall be divided into elements and classified overall, by the highest level element.

2.3 Acceptance and Approval

2.3.1 General Requirements

At the outset of the project/task the contractor shall agree with the customer:

- a Those products (documentation, software and equipments) which require formal acceptance and approval by the customer
- b The procedures for acceptance and approval by the customer

These agreements shall be documented in an Acceptance Plan which shall be prepared by the contractor and subject to customer approval.

For customers external to WHL, other general requirements may be specified in the Terms and Conditions of the business contract.

2.3.2 Documentation

The following is the minimum list of documents which require formal acceptance and approval by external customers. It is not the minimum list of documents required as defined in Section 9.

- a Software Development Plan
- b Software Quality Assurance Plan
- c Acceptance Plan
- d Documentation Plan
- e Software Requirements Specification
- f System/Segment Design Document
- g Interface Requirements Specification
- h Acceptance Test Description

For internal customers, the minimum list of documents should be available either as part of the project documentation, or as defined within Departmental Instructions. In this context, it is the responsibility of the contractor to ensure that the minimum requirements are satisfied.

2.3.3 Equipment and Software

All equipments shall undergo formal acceptance testing as part of their release procedures to the customer.

The documented Acceptance Test Procedure for equipment and software shall be agreed with the customer at minimum one month prior to release.

The results of all formal acceptance testing shall be documented in an Acceptance Test Report.

The contractor shall be responsible for the running of acceptance tests for each software baseline, which should be witnessed by representatives from the customer.

2.4 Project and Software Development Environment

2.4.1 General Requirements

The contractor shall establish a suitable project and software development environment at the outset of the project.

The contractor shall define and document suitable procedures and standards covering the use of these facilities by the project team.

This environment shall be described within the Software Development Plan or relevant Departmental Instruction.

For external contracts the customer should agree the contractors project and software development environment prior to its use on the project.

The contractor shall maintain these facilities under configuration control.

The environment shall provide support for the following software development activities:

- a Software Requirements Analysis
- b Design
- c Coding and Computer Software Unit (CSU) Testing
- d Computer Software Component (CSC) Integration and Testing
- e Computer Software Configuration Item (CSCI) Testing
- f System Integration and Testing
- g Software Development Management (project management)
- h Software Configuration Management and Control
- i Software Quality Assurance

2.4.2 Design Methodologies

The contractor should employ a proven structured design methodology. For example:

- a MASCOT
- b Jackson System Development

2.4.3 Proprietary Design Methodologies

Where possible, design and development, inclusive of more formal methods of structured analysis and design should be employed. (eg: Yourdon Structured Analysis and Design).

2.4.4 Performance and Adequacy

The contractor shall demonstrate the adequacy and correctness of his facilities at an early stage within the project. This shall cover:

- a Host performance
- b Documentation and code maintenance
- c Compiler, assembler and linker performance
- d Target download performance
- e Debugging facilities
- f Test throughput

SDDR-301 Issue 2

2.5 Proprietary Software Management

Proprietary software may be used for two purposes;

- a As a tool for software development
- b As a component of the operational software system

Procedures shall clearly define controls for implementing, testing and acceptance of proprietary software, together with the method of configuration control.

Consideration should be given to the impact of possible errors in proprietary software.

Where proprietary software is used as a component of the operational system it shall be subject to those procedures declared for the development and testing of the operational system.

3 PROJECT PHASES AND SOFTWARE LIFE CYCLE

3.1 Project Phases

The sequential phases of a project are defined as follows:

- a Tendering Phase
- b Development Phase
- c Support Phase

The tendering phase is the period of the project or task up to the agreed commencement of development.

The development phase is the period of a project or task from commencement of development through to delivery of the planned final standard of equipment or software.

The support phase is the period of a project from delivery of the planned final standard of the equipment/software through to the end of the funded support period.

NOTE: Whilst these phases are based on external contracts; similar contractor/customer relationships should be established for in-house software projects/tasks. ie; between or within WHL departments

3.2 Software Development Life Cycle

At the outset of the contract or project the contractor shall define and agree with the customer the equipment or software life cycle model to be used as the basis of his planning and software development management activities. This model shall be documented by the contractor in the Software Development Plan.

The contractor shall adopt the DOD-STD-2167A life cycle as the basic model and tailor it to the specific requirements of the project or task.

This document, in assuming the use of the DOD-STD-2167A life cycle model, is augmented with additional requirements to satisfy WHL's policy for software. Figure 1 depicts the DOD-STD-2167A life cycle model for software development activities.

For information only - Appendix A defines the mapping between DOD-STD-2167A and STARTS life cycles.

The following software development activities are identified by this document:

- a System Requirements Analysis/Design
- b Software Requirements Analysis
- c Preliminary Design
- d Detailed Design
- e Coding and Computer Software Unit (CSU) Testing (Individual module testing)
- f Computer Software Component (CSC) Integration and Testing (Integration and Test of a number of integrated modules performing a specific function)
- g Computer Software Configuration Item (CSCI) Testing (Programme Testing (Target or Host))
- h System Integration and Testing

3.3 Development Support Activities

The following development support activities are identified by this document:

- a Software Development Management (project management)
- b Software Configuration Management and Control
- c Software Quality Assurance

**FIGURE 1.
DOD-STD-2167A System Development Life Cycle.**

4 SOFTWARE DEVELOPMENT ACTIVITIES

4.1 General Requirements

This section defines the software development activities which make up the life cycle, as defined in Section 3. Figure 2 depicts the products from each activity, some of which are deliverable documents. The deliverable documents are more fully described in Section 9.

At the completion of each life cycle phase the software product shall be reviewed to verify it against the requirements established during the previous stage. Work shall not proceed to the next stage until the higher level work has been shown to have been satisfactorily carried out

Independent should be maintained during the verification of the software product. Reviews, Walkthroughs, Inspections and Tests should be carried out by personnel who have had no direct involvement with the production of the item being verified.

4.2 System Requirements Analysis/Design

The objectives of system requirements analysis/design are to:

- a Ensure software requirements of the system specification are consistent, complete
- b Partition system between hardware and software
- c Determine preliminary requirements for each CSCI.
- d Determine preliminary interface requirements for each CSCI.

The detail requirements for this activity are specified in DOD-STD-2167A, Section 5.1. The products of this activity are identified in Figure 2.

4.3 Software Requirements Analysis

The objectives of software requirements analysis are to:

- a Determine complete set of requirements for each CSCI
- b Determine complete set of interface requirements for each CSCI.

The detail requirements for this activity are specified in DOD-STD-2167A, Section 5.2. The products of this activity are identified in Figure 2.

4.4 Preliminary Design

The objectives of preliminary design are to:

- a Allocate software and interface requirements to the CSC's of each CSCI and establish design requirements for each CSCI.
- b Develop preliminary designs for each interface external to each CSCI.
- c Document additional design information for each CSCI.
- d Establish test requirements for conducting CSC integration testing.
- e Determine formal tests to be conducted to verify each CSCI against its requirements.

The detail requirements for this activity are specified in DOD-STD-2167A, Section 5.3. The products of this activity are identified in Figure 2.

SDDR-301 Issue 2

4.5 Detailed Design

The objectives of detailed design are to:

- a Allocate software requirements from the CSC's to the CSU's and establish design requirements for each CSU.
- b Develop the detailed design of the CSCI external interfaces.
- c Document additional design information.
- d Establish test cases and schedules for CSC integration and testing.
- e Establish test requirements, cases and schedules for testing all CSU's.

The detail requirements for this activity are specified in DOD-STD-2167A, Section 5.4. The products of this activity are identified in Figure 2.

4.6 Coding and Computer Software Unit (CSU) Testing

The objectives of coding and CSU testing are to:

- a Develop test procedures for each CSU.
- b Code and test each CSU, verifying the operation and interface of each CSU against its design.
- c Revise and update design documentation and code as necessary.
- d Develop test procedures for conducting each CSC test.

The requirements for this activity are specified in DOD-STD-2167A, Section 5.5. The products of this activity are identified in Figure 2.

The following additional requirements shall also apply:

- e The contractor shall promote the use of High Order Languages for programming, with ADA being the preferred language.
- f The contractor shall have suitable programming and coding standards prior to commencement of work.
- g The contractor shall not use assembler without the written permission of the Head of the relevant Department or delegated authority.
- h The contractor should take all necessary steps to ensure that at entry into service, all equipments shall utilise no more than 50% of their processor and memory capacity.
- i The contractor should provide adequate additional spare capacity to cover growth during the development phase.

NOTE: Items (h) and (i) may be a contractual requirement.

4.7 Computer Software Component (CSC) Integration and Testing

The objectives of CSC integration and testing are to:

- a Conduct all CSC integration and testing to verify the operation and interface of each CSC against its design.
- b Revise and update design documentation and code and re-test as necessary

The detail requirements for this activity are specified in DOD-STD-2167A, Section 5.6. The products of this activity are identified in Figure 2.

4.8 Computer Software Configuration Item (CSCI) Testing

The objectives of CSCI testing are to test each CSCI to verify its operation and interface against its requirements.

The requirements for this activity are specified in DOD-STD-2167A, Section 5.7. The products of this activity are identified in Figure 2.

4.9 System Integration and Testing

The objectives of system integration and testing are to:

- a Integrate each CSCI into the target equipment and verify the operation and interface of the system against its requirements as agreed with the customer.
- b Update and complete all deliverable documentation.

The requirements for this activity are specified in DOD-STD-2167A, Section 5.3. The products of this activity are identified in Figure 2.

FIGURE 2

Software Development Activities and Associated Products

5 SOFTWARE DEVELOPMENT MANAGEMENT ACTIVITIES

5.1 Planning

The contractor should prepare an initial version of the Software Development Plan as part of the tendering phase. The contents of the plan are defined by Section 9.2.1.

The planning activity shall:-

- a Define what work is to be done
- b Define how the work will be done
- c Identify the resources and facilities required to do the work
- d State assumptions or constraints upon which planning is based
- e Define milestones against which progress may be monitored
- f Identify the Software Classification with respect to Usage and Risk (see Section 2.2).

It should be stressed that an essential part of planning is the estimation of effort and elapsed time (schedule) required for the tasks to be conducted. The contractor should include an estimation of schedule based on a formal proven method. The use of the COCOMO method, or typical metrics, or typical metrics which are defined in Section 10, would be desirable. Whichever method is chosen, it shall be declared in the Software Development Plan.

5.2 Reporting

The contractor shall agree and implement with the customer and other organisations as required, a regular progress reporting methodology.

As a minimum, project progress reports should contain the following information:

- a Summary of contract status
- b Summary of programme status
- c Summary of risk status
- d Summary of resource and facility status
- e Summary of milestone status
- f Summary of dependency status
- g Summary of financial status
- h Summary of changes status
- i Summary of project metrics (defined in Section 10.1 to 10.5)
- j Tracked bar chart
- k Actual cumulative spend and accomplishment against planned spend

Reporting should use graphical representation of data, such as Gantt (bar) charts, graphs, etc.

In particular, progress reports shall address deviations from the plan; ie. identifying problems and the actions planned to overcome them.

5.3 Monitoring and Control

The contractor shall implement procedures for internal and subcontractor monitoring and control. These shall be described by or referenced from the Software Quality Assurance Plan.

SDDR-301 Issue 2

5.3.1 Meetings

The contractor shall agree and implement regular progress meetings with the customer and other organisations as required.

As a minimum, these meetings shall address the following:

- a Review of actions
- b Review of project status
- c Review of risks
- d Review of plans
- e Meeting summary

The contractor shall minute all meetings, including records of all actions, and shall distribute minutes to all attendees. If the customer is not represented among the attendees, copies of the minutes shall be sent to the customer.

5.3.2 Control Room/Document

The contractor should set up a project 'control room/document' in which current project data is displayed, allowing easy and informal review of progress and status at any time.

5.4 Subcontracting

Should the need arise, whereby departments within WHL subcontract work to an external agency, then the said subcontractors shall comply with SDDR-301 (this document). The contractor shall be responsible for ensuring that the requirements referenced by and defined within this document are applied to all subcontractors.

The contractor shall be responsible for the satisfactory performance of his subcontractors.

6 SOFTWARE QUALITY ASSURANCE ACTIVITIES

6.1 Planning

The contractor should prepare, for external contracts, an initial version of the Software Quality Plan as part of the tendering phase. For internal projects the Software Quality Plan shall be produced at project commencement. For small internal task developments, the structure shall be identified within a Departmental Instruction. The contents of the plan are defined in Section 9.2.2.

The planning activity shall:

- a Define contractual details, or project details in the case of internal projects.
- b Define responsibilities and identify responsible persons, including project management structure and authorities (see Section 2.1.1).
- c Define the Quality Assurance practices (standards, procedures, methods, etc.,) that shall be used to control the project, including identification of reviews.

6.2 Reporting

The contractor shall report quality assurance activities and metrics as part of the regular progress reporting (see Section 5.2 and 10.5).

6.3 Audits

The contractor shall carry out quality assessments of the project tasks at regular intervals.

The quality assessment should be performed by quality assurance engineers, or software engineers, who are independent of the project phase under assessment.

The schedule for these shall be defined within the Software Development Plan, which shall be referenced by the Software Quality Plan.

The customer reserves the right to carry out its own contractor quality assessments at any time prior to or during the contract. (External contracts only).

Quality assessments shall address the following:

- a Completeness of standards and procedures
- b Compliance of standards and procedures with contract requirements
- c Correctness of standards and procedures
- d Adherence to standards and procedures

The contractor shall produce written reports of all quality assessments, which identify deficiencies found and the corresponding recommended corrective action.

6.4 Reviews, Walkthroughs and Inspections

The contractor shall implement reviews in line with the requirements of DOD-STD-2167A. Useful guide-lines for reviews are also provided by MIL-STD-1521A.

The contractor shall define review and walkthrough procedures based upon the above guide-lines. The procedures shall be implemented on the following activities:

- a System requirements analysis/design
- b Software requirements analysis.
- c Preliminary design
- d Detailed design
- e Coding and CSU testing
- f CSC integration and testing

SDDR-301 Issue 2

- g CSCI testing
- h System integration and testing

The contractor shall arrange for the attendance of appointed representatives of the relevant authorities as required at reviews and walkthroughs.

The contractor shall define and implement inspection procedures for the following:

- i Documentation
- j Source code scrutinies
- k Test/Test report scrutinies

The contractor shall maintain records of **all** reviews, walkthroughs and inspections, identifying the personnel present and their roles.

The customer reserves the right to carry out their own contractor and subcontractors reviews at any time prior to or during the contract.

It is strongly recommended that the customer is encouraged to attend major reviews throughout the project.

6.5 Fault Analyses and Corrective Actions

Whilst the requirement exists for software metrics as declared in Section 10 and configuration (change control) in Section 7, WHL require an analysis of software changes and fault metrics to be undertaken by the contractor. This should include both the identification of the root causes of the non-conformance and the corrective actions to prevent recurrence.

The analysis shall take the form of a report and would be expected to cover all software changes and faults identified in formal deliveries.

It is expected that such analyses forms part of the contractors normal working practice and would not be developed specifically under the contract.

7 SOFTWARE CONFIGURATION MANAGEMENT ACTIVITIES

7.1 Planning

The contractor shall identify requirements for a Software Configuration Management Plan. The contents of the plan or Departmental Instruction are defined by Section 9.2.4. The planning activity shall:

- a Identify items to be placed under configuration control (ie. configurable items)
- b Define procedures for issue and change of configurable items
- c Identify a software library and procedures for controlling access
- d Allocate responsibility to project staff

7.2 Item Identification

All configurable items under the contract shall be allocated a unique item identification reference. This includes the following items:

- a Documentation
- b Software
- c Firmware
- d Hardware

The contractor shall define those items requiring identification and the scheme to be used within the Software Configuration Management Plan.

7.3 Status Accounting and Control

The contractor shall set up and maintain records of all configurable items which detail:

- a Existence
- b Identification
- c Location
- d Status

The contractor shall carry out configuration audits as defined by DOD-STD-2167A

7.4 Issue and Release

The contractor shall define and implement procedures for the issue and release of configurable items at the following stages:

- a Entry into configuration management system
- b Release within the project team
- c Release to the customer

7.5 Software Development Library

The contractor shall set up and maintain a software development library as defined by DOD-STD-2167A containing:

- a Documentation
- b Source code
- c Identification and description of firmware
- d Identification and description of hardware

The library shall cover all forms of storage media (Eg: hardcopy, magnetic, optical).

SDDR-301 Issue 2

The contractor shall define and implement procedures for changing library contents covering the following:

- e Enrolment
- f Loan
- g Up issue
- h Withdrawal and deletion

The contractor shall define and implement back-up procedures to minimise the effects of any disaster (fire, earthquake, etc).

7.6 Software Archive

The contractor shall establish a software archive to hold old versions of configurable items, (no longer required by the development team), required for purposes which include:

- a Backtracking to old configuration items
- b Support of old, but still used configuration items
- c Obsolete, but potentially useful configuration items
- d Audit trails

7.7 Change Control

The contractor shall define and agree with the customer a change control procedure or set of procedures for the following:

- a Contract changes
- b Requirement changes
- c Problem reporting, investigation and problem correction
- d Observation reporting, investigation

7.8 Proprietary Software

The contractor shall define a procedure or set of procedures for the control of:

- a Proprietary Software Library
- b Fault reporting
- c Change/Update Procedure

8 SOFTWARE CLASSIFICATION AND SAFETY ACTIVITIES

8.1 Requirements and Planning

When required by contract, the implementation and planning for the following shall be carried out.

Notwithstanding the contractual requirements, the contractor should perform a preliminary hazard analysis as part of his bid proposal and advise the customer of any elements of the system, subsystem, equipment, or software design, which are considered to be 'essential'. (See Appendix F, Table 1).

Note: For internal projects, where bid proposals are not necessary, a hazard analysis should still be carried out with the above objective maintained (See Section 8.4).

Where any aspect of the contractors design is deemed to have safety implications, the contractor shall prepare a Safety Plan. The contents of the plan are defined by Section 9.2.7.

The planning activity shall:

- a Identify the safety policy
- b Identify design items having safety implications
- c Identify and define safety activities and procedures
- d Allocate responsibility to project staff

8.2 Reporting

The contractor shall report through the regular progress reporting mechanisms the status and achievement of the activities detailed in the Safety Plan.

8.3 Working with the Safety Authority

The contractor shall work with the Safety Authority as required to ensure the required level of visibility is provided and all necessary steps are taken to ensure the safe operation of the software.

8.4 Hazard Analysis

The contractor shall carry out and document a hazard analysis such as defined by interim DEF-STAN 00-56/1.

8.5 Design Requirements

The contractor shall document the design policy, decisions and assumptions relating to the safety aspects of the software.

9 DOCUMENTATION

9.1 General Requirements

For all software, it is desirable that there be a common approach to software development, and a consistency of software documentation. For existing software, the contractor may propose to provide the required information in an alternative manner to reduce costs. These proposed alternatives are subject to customer approval on a case by case basis.

9.1.1 Planning

The contractor should prepare an initial version of the Documentation Plan as part of the tendering phase. For internal projects, the Documentation Plan shall be prepared at project commencement.

As a minimum, the plan shall contain the following information:

- a List of documents to be produced
 - identifying which are deliverable and non-deliverable
 - identifying which Software Development Plan activities they are outputs from
 - identifying which documents require formal acceptance and approval and by whom
- b Specify their purpose, content and format
- c Define documentation hierarchy and relationships
- d Define documents equivalence with this document and with other external requirements for software documentation

9.1.2 Deliverable and Non-deliverable Documentation

For external contracts the contractors, as part of the tendering phase, may be required to provide initial or draft versions of the following software documentation as part of their bid proposals:

- a Software Development Plan
- b Software Quality Assurance Plan
- c Documentation Plan
- d Acceptance Plan

NOTE: This list is for guidance only, as specific contracts may have different tendering requirements.

The contractor shall identify at the tendering phase any information (technical and managerial) which they consider to be company confidential and would not be prepared to make available to external customers.

Software Documents/Topics required by this document are as follows; (documents identified by D are normally considered deliverable; those identified by 'available' are not deliverable, but shall be available to the customer for inspection upon request).

DOCUMENT/TOPICS

Management and Planning Documentation:

Software Development Plan	D
Software Quality Assurance Plan	D
Documentation Plan	D
Software Configuration Management Plan	D
Software Test Plan	D
Acceptance Plan	D
Safety Plan	D
Verification Traceability Matrix	D

Technical Documentation:

System/Segment Design Document	D
Software Requirements Specification	D
Interface Requirements Specification	D
Software Design Document	available
Interface Design Document	available
Software Test Description	available
Software Test Report	available
Acceptance Test Description	D
Acceptance Test Report	D
Software Users Manual	D
Version Description Document	D
Declaration Of Design Performance	D
Software Product Specification	D
Computer System Operators Manual	available
Software Programmers Manual	available
Firmware Support Manual	available
Computer Resources Integrated Support Document	available
Software Development Documents and Procedures Manual	available

9.1.2.1 Reduction or Merger of Deliverables/Documents

For low risk items and subject to agreement with the customer the contractor may reduce the requirements for deliverable documents or merge several deliverables/documents into one.

9.1.2.2 'Electronic' Delivery

The contractor will, with the agreement of the customer, deliver documentation in an 'electronic' form; ie. in machine readable format conforming to either PC/DOS or VAX/VMS systems.

9.1.3 Maintenance and Updates during Development

The contractor whilst maintaining all documentation under configuration control shall update all copies of the following documents to the latest issue as agreed with the customer:

- a All planning documentation
- b All requirements specifications
- c Acceptance test descriptions and reports

9.1.4 Software Development Files

The requirements for software development files are specified in DOD-STD-2167A.

The scope of the software development files shall be extended to cover the following additional requirements

- a Requirements documentation and data
- b Traceability and cross referencing information
- c Review, walkthrough and inspection records
- d Amendment histories
- e Software metrics information

SDDR-301 Issue 2

9.2 Management and Planning Documentation

9.2.1 Software Development Plan

The format and content of the plan shall be as defined by DID DI-MCCR-80030 with the inclusion of the following additional information:

- | | | |
|---|---|-----------------------------------|
| a | Assumptions, Constraints and Dependencies | (to be included as Section 1.5) |
| b | Software Classification | (to be included as Section 1.6) |
| c | Planned staffing profile | (to be included as Section 3.1.5) |
| d | Bar/Gantt chart | (to be included as Section 3.2.4) |
| e | Planned cumulative spend profile | (to be included as Section 3.2.5) |
| f | Baseline software metrics | (to be included as Section 4.4) |

Effective Risk Management is seen as a particularly important aspect of the overall software development activity. Section 4.3 of the Plan should include sections on Risk Assessment (Identification, Analysis and Prioritisation) and Risk Management (Planning, Execution, Monitoring and Control). A typical Risk Monitoring and Control activity would be a regular review of a "Top Ten" Risk Items list.

9.2.2 Software Quality Assurance Plan

The contents and format of this document shall be as defined by the contents list at Appendix D1.

9.2.3 Documentation Plan

The contents and format of this document shall be as defined by the contents list at Appendix D2.

9.2.4 Software Configuration Management Plan

This document shall describe the Software Configuration Management organisation, policies and procedures which will apply throughout the entire software life cycle. It shall define a systematic approach for the identification and release of software and associated documentation including the evaluation, co-ordination, approval, implementation and recording of all changes.

The contents and format of this document shall be as defined by the contents list at Appendix D3.

9.2.5 Software Test Plan

The contents and format of this document shall be as defined by DID DI-MCCR-80014.

9.2.6 Acceptance Plan

The contents and format of this document shall be as defined by the contents list at Appendix D4.

9.2.7 Safety Plan

The contents and format of this document shall be as defined by the contents list at Appendix D5.

NOTE: For external contracts ONLY, the documents referred to in paras 9.2.1 to 9.2.7 inclusive, alternative contents and formats shall only be permitted by written consent of the customer.

9.2.8 Verification Traceability Matrix

This document will be required for 'essential' software. The document shall contain the following sections:

- Verification Status Matrix
- Test Traceability Matrix

9.2.8.1 Verification Status Matrix

This section will identify the status of the verification activities applicable to each item of documentation and software. This includes design reviews, specification scrutinies, code scrutinies, tests and test scrutinies.

9.2.8.2 Test Traceability Matrix

This section will consist of a cross reference matrix identifying each explicit or implicit requirement and the test case or cases which are asserted to demonstrate compliance.

9.3 Technical Documentation

9.3.1 Requirements Documentation

The contractor shall agree with the customer the required requirements documentation for the contract.

The format and contents shall be based on the following DID's:

System/Segment Design Document	DI-CMAN-80534
Software Requirements Specification	DI-MCCR-80025
Interface Requirements Specification	DI-MCCR-80026

9.3.2 Design Documentation

The contractor shall agree with the customer the required design documentation for the contract.

The format and contents shall be based on the following DID's:

Software Design Document	DI-MCCR-80012
Interface Design Document	DI-MCCR-80027

9.3.3 Test Documentation

The contractor shall agree with the customer the required test documentation for the contract.

The format and contents shall be based on the following DID's:

Software Test Description	DI-MCCR-80015
Software Test Report	DI-MCCR-80017

9.3.4 User Documentation

The contractor shall agree with the customer the required user documentation for the contract.

The format and contents shall be based on the following DID:

Software Users Manual	DI-MCCR-80019
-----------------------	---------------

SDDR-301 Issue 2

9.3.5 Issue and Release Documentation

The contractor shall agree with the customer the required issue and release documentation for the contract.

The format and contents shall be based on the following:

Version Description Document	DI-MCCR-80013
Declaration of Design Performance (DDP)	see note

NOTE: The requirement for a DDP will be contract dependant.

The contractor shall agree with the customer the required issue and release documentation for Instrumentation Software required to be fitted to aircraft. See Appendix D6.

9.3.6 Maintenance and Support Documentation

The contractor shall agree with the customer the required maintenance and support documentation for the contract.

The format and contents shall be based on the following DID's:

Software Product Specification	DI-MCCR-80029
Computer System Operators Manual	DI-MCCR-80018
Software Programmers Manual	DI-MCCR-80021
Firmware Support Manual	DI-MCCR-80022
Computer Resources Integrated Support Document	DI-MCCR-80024

9.3.7 Software Development Documents and Procedures Manual

This document or documents shall define the relevant procedures, standards or code of practices to be used by the contractor during the development of software and shall be used as a reference manual by the software development team.

It is expected that such document/documents would exist as part of the contractors normal working practice and would not be developed specifically under the contract.

9.4 Documentation Equivalence Tables

The relationship between the documentation requirements of this document and the following standards is defined in Table 9.1.

- a DOD-STD-2167A
- b JSP 188 Ed 4

Table 9.1

Documentation Equivalence Table

		DOD-STD-2167A	JSP 188 Ed 4
1	Software Dev Plan	Software Dev Plan	System Software Project Management
2	Software QA Plan	-	System Software Spec Quality Assurance
3	Documentation Plan	-	-
4	Software Configuration Management Plan	-	-
5	Software Test Plan	Software Test Plan	Test Plan
6	Acceptance Plan	-	-
7	Safety Plan	-	-
8	Verification Traceability Matrix	-	-
9	System/Segment Design Document	System/Segment Design Document	System Software Specification
10	Software Requirements Specification	Software Requirements Specification	System Software Specification
11	Interface Requirements Specification	Interface Requirements Specification	System Software Specification
12	Software Design Document	Software Design Document	Design Documentation
13	Interface Design Document	Interface Design Document	-
14	Software Test Description	Software Test Description	Test Case and Test Procedures Specification
15	Software Test Report	Software Test Report	-
16	Acceptance Test Procedure	-	Acceptance
17	Acceptance Test Report	-	Acceptance

Table 9.1 (cont)

Documentation Equivalence Table

		DOD-STD-2167A	JSP 188 Ed 4
18	Version Description Document	Version Description Document	-
19	Software Product Specification	Software Product Specification	Basic Coded Unit Documentation
20	Computer System Operators Manual	Computer System Operators Manual	In-Service Maintenance
21	Software Users Manual	Software Users Manual	In-Service Maintenance Manual
22	Software Programmers Manual	Software Programmers Manual	In-Service Maintenance
23	Firmware Support Manual	Firmware Support Manual	-
24	Computer Resources Integrated Support Document	Computer Resources Integrated Support Document	-
25	Declaration Of Design & Performance	-	-
26	Software Development Standards and Procedures Manual	-	-

10 SOFTWARE METRICS

10.1 General Requirements

At the outset of an external project, the contractor shall agree with the customer the definitions and standards to be used in estimating and measuring software metrics. The software metrics defined in this section are a comprehensive set covering all aspects of the development programme and relating to the use of COCOMO for estimation. The customer shall agree with the contractor a reasonable subset of these metrics to be collected throughout the software development programme.

The contractor shall estimate management, size and performance, reliability and quality software metrics as part of the bid proposal. The contractor shall measure and re-estimate these software metrics throughout the duration of the contract.

The contractor shall report all metrics to the customer as defined by the reporting procedures given in Sections 5, 6 and 8.

The contractor shall maintain records of all metrics throughout the duration of the contract.

The contractor shall calculate the variance associated with all metrics.

The contractor shall be responsible for ensuring the accuracy of his estimates within the specified variance.

The contractor shall use graphical means of recording and displaying project metrics wherever possible.

For internal projects the contractor shall identify the methods utilised and, where possible, move towards more formal methods of utilising metrics appropriate to the project/task. NOTE: The requirement for software metrics will be dependant on:

- a The contract requirements
- b The size of the project
- c The Software Classification (see para., 2.2).

10.2 Typical Management Metrics

a Project attributes

- i) Software personnel attributes
 - engineer capability
 - engineer experience
 - applications experience
 - virtual machine experience
 - programming language experience
 - methods and tools experience
- ii) Computer attributes
 - execution time constraint
 - main storage constraint
 - virtual machine volatility
 - computer turn round time
 - programme load/download time

SDDR-301 Issue 2

iii) Development environment attributes

- modern practices
- use of software tools
- required development schedule
- working conditions
- maturity of tools and methods
- average size of project team
- peak size of project team
- project cost breakdown structure

b Development process indicators

- number of functions designed, implemented integrated and tested, delivered
- number of lines of code designed, implemented, integrated and tested, delivered
- effort expended on requirements, design, implementation, integration and test
- estimate to complete requirements, design, implementation, integration and test
- number of 'modules' designed, implemented, integrated and tested
- number of 'units' designed, implemented, integrated and tested
- number of 'design points' completed and/or to complete
- number of 'test points' passed/failed/not tried
- actual productivity versus planned productivity

c Test progress indicators

- number of units/modules tested
- number of units/modules integrated
- number of 'integration points' passed/failed/not tried
- number of 'test points' passed/failed/not tried

d Software volatility indicators

- number of observations raised
- number of observations closed
- number of faults raised
- number of faults rejected
- number of corrections pending
- number of corrections in-progress
- number of corrections completed
- number of changes raised
- number of changes rejected
- number of changes pending
- number of changes in-progress
- number of changes completed
- rate of change of function point estimate
- rate of change of lines of code estimate

e Incremental release content indicators

- increment/total function point estimate ratio
- increment/total lines of code estimate ratio

f Software productivity indicators

- number of developed lines of code per manday
- number of delivered lines of code per manday
- number of delivered function points per manday
- number of developed lines of code per elapsed day
- number of delivered lines of code per elapsed day
- number of delivered function points per elapsed day

10.3 Typical Size and Performance Metrics

The contractor should estimate and measure the following size and performance software metrics:

a Software size indicators

- function point
- lines of code
- effort
- number of 'modules'
- number of 'units'
- number of 'design points'
- number of 'test points'

b Computer resource utilisation indicators

- estimated memory utilisation (RAM and ROM)
- actual memory utilisation (RAM and ROM)
- estimated processor utilisation
- variance on estimated processor utilisation
- actual processor utilisation
- estimated response times
- variance on estimated response times
- actual response times

10.4 Typical Reliability Metrics

The contractor should record and produce the following software reliability metrics:

a Software reliability indicators

- Number of faults found per 1000 lines of code
- Number of faults found per function point
- Number of faults found per per unit elapsed time
- Number of faults found per per unit operating time

The contractor should analyse and record:

- where faults/observations were found (ie in which software modules)
- the life cycle activity during which the faults/observations were found
- the life cycle activity in which the faults are thought to have been introduced into the development

b Reliability Growth Metrics and Prediction Of Final System Maturity

The contractor should report the reliability growth in terms of the total number of faults found against time.

The reliability growth curve should be used to predict the final system maturity.

10.5 Typical Quality Metrics

The contractor should estimate and measure the following software quality metrics:

a Audits

- number of audits planned
- number of audits carried out
- number of audit actions raised
- number of audit actions closed

SDDR-301 Issue 2

b Reviews

- number of reviews planned
- number of review carried out
- number of review actions raised
- number of review actions closed

c Walkthroughs

- number of walkthroughs planned
- number of walkthroughs carried out
- number of walkthrough actions raised
- number of walkthrough actions closed

d Inspections

- number of inspections planned
- number of inspections carried out
- number of inspection actions raised
- number of inspection actions closed

11 TAILORING OF THESE REQUIREMENTS FOR EXISTING EQUIPMENTS AND SOFTWARE

11.1 Establishing Level of Software Reuse

The contractor shall estimate and define as part of the bid proposal the level of reuse of existing software. This shall be expressed as a percentage where:

- a 100% => no new software
- b 0% => all new software

11.2 Minimum Mandatory Requirements

The following list identifies which sections of this document are mandatory and shall apply to all contractors, and which sections may be tailored, by mutual agreement of the customer and the contractor, to meet the specific requirements of the contract :

Section	Mandatory/Tailorable
1 INTRODUCTION	
1.1 Purpose	Mandatory
1.2 Scope	Mandatory
1.2.1 Applicability	Mandatory
1.2.2 Superseded Standards	Mandatory
1.3 Applicable Standards	Mandatory
1.4 Other Related Standards	Mandatory
1.5 Associated Policies and Documents	Mandatory
1.6 Interpretation	Mandatory
1.7 Control and Use of Document	Mandatory
1.8 Overview of Contents	Mandatory
2 GENERAL REQUIREMENTS	
2.1 Roles and Responsibilities	
2.1.1 Authorities	Mandatory
2.1.2 Customer	Mandatory
2.1.3 Contractor	Mandatory
2.1.4 Contract	Mandatory
2.1.5 Project	Mandatory
2.1.6 Task	Mandatory
2.2 Software Classification:Usage and Risk	Mandatory
2.3 Acceptance and Approval	
2.3.1 General Requirements	Mandatory
2.3.2 Documentation	Tailorable
2.3.3 Equipment and Software	Mandatory
2.4 Project and Software Development Environments	
2.4.1 General Requirements	Mandatory
2.4.2 Design Methodologies	Tailorable
2.4.3 Proprietary Design Methodologies	Tailorable
2.4.4 Performance and Adequacy	Mandatory
2.5 Proprietary Software Management	Mandatory

3	PROJECT PHASES AND SOFTWARE LIFE CYCLE	
3.1	Project Phases	Tailable
3.2	Software Development Life Cycle	Tailable
3.3	Development Support Activities	Tailable
4	SOFTWARE DEVELOPMENT ACTIVITIES	
4.1	General Requirements	Mandatory
4.2	System Requirements Analysis/Design	Tailable
4.3	Software Requirements Analysis	Tailable
4.4	Preliminary Design	Tailable
4.5	Detailed Design	Tailable
4.6	Coding and Computer Software Unit (CSU) Testing	Tailable
4.7	Computer Software Component (CSC) Integration and Testing	Tailable
4.8	Computer Software Configuration Item (CSCI) Testing	Tailable
4.9	System Integration and Testing	Tailable
5	SOFTWARE DEVELOPMENT MANAGEMENT ACTIVITIES	
5.1	Planning	Mandatory
5.2	Reporting	Mandatory
5.3	Monitoring and Control	
5.3.1	Meetings	Mandatory
5.3.2	Control Room/Document	Tailable
5.4	Subcontracting	Mandatory
6	SOFTWARE QUALITY ASSURANCE ACTIVITIES	
6.1	Planning	Mandatory
6.2	Reporting	Tailable
6.3	Audits	Tailable
6.4	Reviews, Walkthroughs and Inspections	Tailable
6.5	Fault Analyses and Corrective Actions	Mandatory
7	SOFTWARE CONFIGURATION MANAGEMENT ACTIVITIES	
7.1	Planning	Mandatory
7.2	Item Identification	Mandatory
7.3	Status Accounting and Control	Tailable
7.4	Issue and Release	Tailable
7.5	Software Development Library	Tailable
7.6	Software Archive	Mandatory
7.7	Change Control	Mandatory
7.8	Proprietary Software	Mandatory
8	SOFTWARE CRITICALITY AND SAFETY ACTIVITIES	
8.1	Requirements and Planning	Mandatory
8.2	Reporting	Tailable
8.3	Working with the Safety Authority	Tailable
8.4	Hazard Analysis	Tailable
8.5	Design Requirements	Tailable

9 DOCUMENTATION

9.1	General Requirements	
9.1.1	Planning	Mandatory
9.1.2	Deliverable and Non-deliverable Documentation	Tailable
9.1.2.1	Reduction Or Merger Of Deliverables	Tailable
9.1.2.2	'Electronic' Delivery	Tailable
9.1.3	Maintenance and Updates during Development	Mandatory
9.1.4	Software Development Files	Tailable
9.2	Management and Planning Documentation	
9.2.1	Software Development Plan	Tailable
9.2.2	Software Quality Assurance Plan	Tailable
9.2.3	Documentation Plan	Tailable
9.2.4	Software Configuration Management Plan	Tailable
9.2.5	Software Test Plan	Tailable
9.2.6	Acceptance Plan	Tailable
9.2.7	Safety Plan	Tailable
9.2.8	Verification Traceability Matrix	Tailable
9.2.8.1	Verification Status Matrix	Tailable
9.2.8.2	Test Traceability Matrix	Tailable
9.3	Technical Documentation	
9.3.1	Requirements Documentation	Tailable
9.3.2	Design Documentation	Tailable
9.3.3	Test Documentation	Tailable
9.3.4	User Documentation	Tailable
9.3.5	Issue and Release Documentation	Tailable
9.3.6	Maintenance and Support Documentation	Tailable
9.3.7	Software Development Standards and Procedures Manual	Tailable
9.4	Documentation Equivalence Tables	

10 SOFTWARE METRICS

10.1	General Requirements	Mandatory
10.2	Typical Management Metrics	Tailable
10.3	Typical Size and Performance Metrics	Tailable
10.4	Typical Reliability Metrics	Tailable
10.5	Typical Quality Metrics	Tailable

11 TAILORING OF THESE REQUIREMENTS FOR EXISTING EQUIPMENTS AND SOFTWARE

11.1	Establishing Level of Software Re-use	Mandatory
11.2	Minimum Mandatory Requirements	Mandatory

DOD-STD-2167A and STARTS Life Cycle Equivalence Table

The following Table identifies the equivalence between phases of the DOD-STD-2167A Life Cycle and the STARTS Life Cycle. It should be noted that whilst a system life cycle is comprised of both hardware and software life cycle phases, only those relevant to software are addressed here.

APPENDIX B

Glossary

The following are taken from the list of definitions provided by DOD-STD-2167A, but includes enhancements.

Computer Data Definition: A statement of the characteristics of the basic elements of information operated upon by hardware in responding to computer instructions. These characteristics may include, but are not limited to, type, range, structure, and value.

Computer Hardware: Devices capable of accepting and storing computer data, executing a systematic sequence of operation on computer data, or producing control outputs. Such devices can perform substantial interpretation, computation, communication, control, or other logical functions.

Computer Resources: The totality of computer hardware, software, personnel, documentation, supplies and services applied to a given effort.

Computer Software (or software): A combination of associated computer instructions and computer data definitions required to enable the computer hardware to perform computational or control functions. It includes operating systems, compilers, test routines, as well as the associated documents such as, specifications, test plans, test results and user instructions.

Computer Software Component (CSC): A distinct part of a computer software configuration item (CSCI). CSCs may be further decomposed into other CSCs and Computer Software Units (CSUs). (Integrated modules performing a specific function.)

Computer Software Configuration Item (CSCI): A configuration item for computer software. (Programme testing of a series of integrated modules (Target or Host))

Computer Software Documentation: Technical data or information, including computer listings and printouts, which documents the requirements, design, or details of computer software, explains the capabilities and limitations of the software, or provides operating instructions for using or supporting computer software during the software's operational life.

Computer Software Unit (CSU): An element (Module or Sub-Module) specified in the design of a Computer Software Component (CSC) that is separately testable.

Development Configuration: The contractor's software and associated technical documentation that defines the evolving configuration of a CSCI during development. It is under the development contractor's configuration control and describes the software design and implementation. The Developmental Configuration for a CSCI consists of a Software Design Document and source code listings. Any item of the Development Configuration may be stored on electronic media.

Evaluation: The process of determining whether an item or activity meets specified criteria.

Firmware: The combination of a hardware device and computer instructions or computer data that reside as read-only software on the hardware device.

Formal Qualification Testing (FQT): A process that allows the contracting agency to determine whether a configuration item complies with the allocated requirements for that item.

Independent Verification and Validation (IV&V): Verification and validation performed by a contractor or agency that is not responsible for developing the product or performing the activity being evaluated. IV&V is an activity that is conducted separately from the software development activities governed by this document.

Release: A configuration management action whereby a particular version of software is made available for a specific purposes (Eg released for test).

Reusable Software: Software developed in response to the requirements for one application that can be used, in whole or in part, to satisfy the requirements of another application.

Software Development File (SDF): A repository for a collection of material pertinent to the development or support of software. Contents typically include (either directly or by reference) design considerations and constraints, design documentation and data, schedule and status information, test requirements, test cases, test procedures, and test results.

Software Development Library (SDL): A controlled collection of software, documentation, and associated tools and procedures used to facilitate the orderly development and subsequent support of software. The SDL includes the Development Configuration as part of its contents. A software development library provides storage of and controlled access to software and documentation in human-readable form, machine-readable form, or both. The library may also contain management data pertinent to the software development project.

Software Engineering Environment: The set of automated tools, firmware devices, and hardware necessary to perform the software engineering effort. The automated tools may include but are not limited to compilers, assemblers, linkers, loaders, operating system, debuggers, simulators, emulators, test tools, documentation tools, and data base management system(s).

Software Support: The sum of all activities that take place to ensure that implemented and fielded software continues to fully support the operational mission of the software.

Software Test Environment: A set of automated tools, firmware devices, and hardware necessary to test software. The automated tools may include but are not limited to test tools such as simulation software, code analyzers, etc. This may also include those tools used in the software engineering environment.

System Specification: A system level requirements specification. A system specification may be a System/Segment Specification (SSS), Prime Item Development Specification (PIDS), or Critical Item Development Specification (CIDS).

Validation: The process of evaluating software to determine compliance with specified requirements.

Verification: The process of evaluating the products of a given software development activity to determine correctness and consistency with respect to the products and standards provided as input to that activity.

APPENDIX B (cont)

Version: An identified and documented body of software. Modifications to a version of software (resulting in a new version) require configuration management actions by either the contractor, the contracting agency, or both.

SDDR-301 Issue 2
APPENDIX C

Abbreviations

CDR	Critical Design Review
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSU	Computer Software Unit
DID	Data Item Description
DOD	Department of Defense (US)
FQR	Formal Qualification Review
FQT	Formal Qualification Testing
FSM	Firmware Support Manual
HOL	High Order Language
ISO	International Organization for Standardization
ITT	Invitation To Tender
LRU	Line Replaceable Unit
MOD	Ministry of Defence
PCA	Physical Configuration Audit
PDR	Preliminary Design Review
SDD	Software Design Document
SDF	Software Development File
SDR	System Design Review
SSR	Software Specification Review
TRR	Test Readiness Review
VV&T	Verification, Validation & Test
WHL	Westland Helicopters Limited

APPENDIX D1**SOFTWARE QUALITY ASSURANCE PLAN****Contents****1.0 INTRODUCTION**

- 1.1 Purpose
(The purpose of the document)
- 1.2 Scope
(A high level description of what equipment and software the document is applicable to and its position in relation to other baselines)
- 1.3 Control of Document
(A description of how the document will be controlled and the procedure for making changes)

2.0 APPLICABLE DOCUMENTS**3.0 GLOSSARY OF TERMS & ABBREVIATIONS****4.0 MANAGEMENT**

(The management responsibilities for the project, supplemented by an organisational chart showing lines of responsibility. It should also indicate how progress shall be monitored and reported to the customer, particularly with respect to processor loading and memory)

5.0 AUTHORITIES

(Identify the applicability, responsibility and interfaces for the following authorities:

- System Design Authority
- System Verification, Validation & Test Authority
- Acceptance Authority
- Safety Authority)

6.0 STANDARDS, PRACTICES AND CONVENTIONS

- 6.1 Analysis, Design and Programming Standards, Practices and Conventions.
(Identify the standards, practices and conventions to be applied. The subjects covered should include the basic technical, design and programming activities involved, such as documentation naming and coding, programming languages and unit testing)
- 6.2 Compliancy.
(State how compliancy with the standards, practices and conventions defined in the preceding section shall be monitored and assured at each phase of the development life cycle.)

7.0 DOCUMENTATION

(Identify and detail the document 'types' governing the development and verification of the software and clearly state how they are checked for adequacy and approved. Note that this section is not concerned with documentation delivery schedules. A complete description of documentation for the project or task should be detailed in the Documentation Plan).

8.0 REVIEWS & AUDIT

(This section should specify all the 'types' of reviews/audit to be adopted for the development of the project that ensure that the identified technical and procedural requirements are adhered to. As a minimum the following information should be clearly stated for each review type:

- the type of review/audit
- who calls the review/audit
- who attends
- who chairs the meeting
- what form does the agenda take
- what are the inputs to the review/audit
- what are the outputs
- what check-list(s) are used
- how non-compliances to the life cycle are reported

Where appropriate, references to relevant company procedures (which would be listed in Section 6.0) should be given. Note that the customer reserves the right to attend all major design reviews).

9.0 SOFTWARE CONFIGURATION MANAGEMENT

(This section should give a brief summary of the methods to be adopted for configuration management. A complete description of the configuration management for the project should be detailed in the Software Configuration Management Plan).

10.0 APPROVAL SIGNATORIES

(Identify all approved signatories. It should include those for test reports, documents, certificates of conformance, concessions etc)

11.0 TOOLS, TECHNIQUES AND METHODOLOGIES

(This section should identify the tools, techniques, methodologies and records to be employed in the performance of the work which will support Quality Assurance objectives and describe how their use will augment or satisfy Quality Assurance requirements. Examples include system analysis techniques, functional and performance requirements analysis, error analysis, specification tracing and coding conventions)

12.0 CONCESSION PROCEDURE

(Details of the concession procedure should be specified)

13.0 QUALITY ASSURANCE ACTIVITIES

(This section should detail both the activity and the intensity of the Quality Assurance activity at the various stages of the project/task. It should include but not be limited to, Audits, Design Reviews, Trials and Testing. The activity can be carried out by a software engineer who has knowledge of, but is not directly involved with the development of the project or task.)

14.0 AGREED NON-COMPLIANCE WITH SDDR-301. (This document.)

(Any non-compliance with the requirements of this document identified at the tendering phase or project outset should be declared. This is particularly important where WHL sub-contract work against this document.)

15.0 REVISION OF THE QUALITY PLAN

(This section should include a statement of the intention to review the plan. It should identify at what stages in the project the plan will be reviewed)

APPENDIX D2**DOCUMENTATION PLAN****Contents**

- 1.0 INTRODUCTION
 - 1.1 Purpose
(The purpose of the document)
 - 1.2 Scope
(A high level description of what equipment and software the document is applicable and its position in relation to other baselines)
 - 1.3 Control of Document
(A description of how the document will be controlled and the procedures for making changes)
- 2.0 GLOSSARY OF TERMS & ABBREVIATIONS
- 3.0 IDENTIFICATION OF DOCUMENTATION
(List of all documents to be produced, including:
 - titles
 - reference numbers
 - approval authorities
 - purpose of document
 - outline of contents
 - identify life cycle activity in which individual documents are to be produced
 - identify which documents are deemed deliverable and hence require acceptance by the customer)
- 4.0 RELATIONSHIP OF DOCUMENTATION
(Identification of relationships between all documentation, including, where appropriate, the document hierarchy.)
- 5.0 RELEVANT STANDARDS
(Identification of standards relevant to production of documents, including identification of styles, etc., to be employed.)
- 6.0 PROCEDURES FOR ISSUING DOCUMENTS
 - 6.1 Procedures for Issuing Baseline Documents
 - 6.2 Procedures for Maintaining and Re-issuing Documents

SOFTWARE CONFIGURATION MANAGEMENT PLAN

Contents

- 1.0 INTRODUCTION
 - 1.1 Purpose
(The purpose of the document)
 - 1.2 Scope
(A high level description of what equipment and software the document is applicable and its position in relation to other baselines)
 - 1.3 Control of Document
(A description of how the document will be controlled and the procedures for making changes)
- 2.0 APPLICABLE DOCUMENTS
- 3.0 GLOSSARY OF TERMS & ABBREVIATIONS
- 4.0 ORGANISATION AND RESPONSIBILITIES
 - 4.1 Organisation
(State all the organisations/groups/departments etc. concerned with the software configuration management and their associated responsibilities).
 - 4.2 Software Development Library
(An overview/description of the software development library (or libraries) to be used on the project).
- 5.0 IDENTIFICATION OF SOFTWARE ITEMS
(Define all types of software related items (LRU, source files, object files, listing files, build files, software executables, documentation, test cases, support software, hardware etc.) to be configured and clearly state how they are identified).
- 6.0 ACCEPTANCE OF ITEMS INTO THE SOFTWARE DEVELOPMENT LIBRARY
(For each configuration item listed in para 5.0 a description is needed of the prerequisites/checking to be performed before the item can be accepted and lodged in the software development library).
- 7.0 RELEASE OF ITEMS FROM THE SOFTWARE DEVELOPMENT LIBRARY
(The rules and procedures for the release of each configuration item, as listed in para 5.0, from the software development library. This to include the procedures used to build/release a version of the software to the customer).
- 8.0 CHANGE CONTROL
(A description of the change control procedures for all of the configuration items as listed in para 5.0).
- 9.0 SOFTWARE CONFIGURATION STATUS ACCOUNTING
(The method of determining the current configuration status of any configured item to give progress information, management visibility, traceability and history of changes).
- 10.0 SOFTWARE CONFIGURATION MANAGEMENT TOOLS
(A description of all of the configuration management tools to be adopted).
- 11.0 MEDIA CONTROL
(State the methods and facilities to be used to protect computer program media from unauthorised access or inadvertent damage or degradation).

APPENDIX D4**ACCEPTANCE PLAN****Contents**

- 1.0 INTRODUCTION
- 1.1 Purpose
(The purpose of the document)
- 1.2 Scope
(A high level description of what equipment and software the document is applicable to and its position in relation to other baselines)
- 1.3 Control of Document
(A description of how the document will be controlled and the procedures for making changes)
- 2.0 APPLICABLE DOCUMENTS
- 3.0 GLOSSARY OF TERMS AND ABBREVIATIONS
- 4.0 ITEMS REQUIRING ACCEPTANCE BY WHL
(A list of the items requiring acceptance by WHL eg Documents, Software, Equipment)
- 5.0 ACCEPTANCE CRITERIA
(The criteria by which the items listed in 4.0 will be assessed acceptable eg
- | | |
|--------------------------------|----------------------------|
| Documentation: | Verification status |
| | Issue status |
| | Format |
| | Consistency |
| | Traceability |
| Software: | Acceptance testing |
| | Verification status |
| | Pre-release audit |
| | Pre-release design review |
| Equipment (embedded software): | Acceptance testing |
| | Verification status |
| | Pre-release audit |
| | Pre-release design review) |
- 6.0 ACCEPTANCE METHODS
(The methods by which the items listed in 4.0 will be Assessed acceptable eg
- Testing Acceptance testing conducted by the supplier and witnessed by WHL.
 - Review Review of design and test documentation.
 - Audit Functional and physical configuration audits performed on the deliverable products.
 - Demonstration A demonstration by the supplier of particular aspects of the deliverable product Eg.
 - Performance
 - Man machine interface)

SDDR-301 Issue 2
APPENDIX D4 (CONT)

7.0 RESPONSIBILITIES

(A statement of the responsibilities of the supplier and WHL regarding:

- Acceptance activities at the suppliers' site
- Acceptance activities at WHL
- Provision of facilities
- Witnessing of tests
- Attendance of reviews
- Response to Documentation.)

APPENDIX D5

SAFETY PLAN

Contents

- 1.0 INTRODUCTION
 - 1.1 Purpose
(The purpose of the document)
 - 1.2 Scope
(A high level description of what equipment and software the document is applicable and its position in relation to other baselines)
 - 1.3 Control of Document
(A description of how the document will be controlled and the procedures for making changes)
- 2.0 APPLICABLE DOCUMENTS
- 3.0 GLOSSARY OF TERMS AND ABBREVIATIONS
- 4.0 SAFETY POLICY
- 5.0 DESIGN POLICY AND APPROACH
- 6.0 REQUIREMENTS FOR VV&T ACTIVITIES TO ENSURE SAFETY
- 7.0 ROLES AND RESPONSIBILITIES WITH RESPECT TO SAFETY
- 8.0 PLANNED ACTIVITIES WITH RESPECT TO ACHIEVING THE SAFETY REQUIREMENTS
- 9.0 ASSUMPTIONS AND RESTRICTIONS

INSTRUMENTATION SOFTWARE

It is a requirement that all software developed for the validation of equipment/systems for WHL is developed in accordance with this document. A disciplined approach for its specification, design, implementation and verification should be adopted and that the total development process is fully documented.

The exact documentation set for instrumentation software will be dictated by the assessed classification of the equipment/system under validation and the interface between the equipment/system and validation software. i.e. can the instrumentation software in any way affect the system under assessment?

A full set of documentation approved by WHL should be in place prior to fitment to aircraft.

APPENDIX F**TABLES FOR CLASSIFICATION OF SOFTWARE**

The extent of tailoring of this document to a reduced set of requirements for software development shall be agreed between the contractor and the customer on the basis of the software Usage and Risk associated with the particular development programme. See Tables 1 and 2.

Table 1**Software Classification-Usage**

The overall usage shall be assessed according to the following classifications:

Failure Effect	Classification
Occurrence of any failure condition or design error would significantly reduce the capability of the SYSTEM/SUBSYSTEM, or the ability of the OPERATOR/USER to cope with any adverse operating conditions.	ESSENTIAL
Occurrence of any failure condition or design error would not significantly reduce the SYSTEM/SUBSYSTEM capability or OPERATOR/USER ability to cope with any adverse operating conditions.	NON ESSENTIAL

Table 2
Software Classification-Risk

The overall risk shall be assessed according to the following classifications:

1 Technical Innovation

This shall be assessed on the basis of percentage of new technology to the contractor (ie hardware or software technology of which the contractor has had no prior experience). Less new technology implies lower risk.

0 to 25% new		0	lo risk
26 to 50% new		1	
51 to 75% new		2	
76 to 100% new		3	hi risk

2 Technical Complexity

This shall be assessed on the basis of ratio of function count to estimated lines of code. Fewer lines of code per function shall imply greater risk.

small ratio	Eg <0.01	0	lo risk
	Eg <0.1	1	
	Eg <0.5	2	
large ratio	Eg >0.5	3	hi risk

3 Schedule

This shall be assessed against the equivalent COCOMO based schedule derived for the programme. A shorter schedule shall imply higher risk.

longer than COCOMO schedule		0	lo risk
equal to COCOMO schedule		1	
75% to 99% of COCOMO schedule		2	
1% to 74% of COCOMO schedule		3	hi risk

4 Project Size

This shall be assessed on the basis of the number of people on the development team. Fewer people shall imply less risk.

1 to 5		0	lo risk
6 to 10		1	
11 to 25		2	
>25		3	hi risk

5 Importance to Envisaged Programme

This shall be assessed against the effect of any delay in the development programme to the overall aircraft programme. Lower effect to the overall programme shall imply less risk.

no effect on aircraft programme		0	lo risk
slight effect on aircraft programme		1	
serious effect on aircraft programme		2	
aircraft programme totally dependent		3	hi risk

APPENDIX F (Cont)**6 Level of Reuse of Existing Software**

This shall be assessed on the basis of the percentage of existing software that shall be reused.
Greater reuse of existing software shall imply less risk.

76 to 100% reuse	0	lo risk
51 to 75% reuse	1	
26 to 50% reuse	2	
0 to 25% reuse	3	hi risk

7 Calculation of Overall Risk

The overall level of risk shall be assessed by summing the risk levels scored against steps 1 to 6 above.

ie. Total risk score = sum of individual risk scores for steps 1 to 6.

Total Risk Score	Classification
0-3	LOW risk
4-8	MINOR risk
9-13	MODERATE risk
14-18	MAJOR risk