NATO STANDARD

AQAP-2210

NATO SUPPLEMENTARY SOFTWARE QUALITY ASSURANCE REQUIREMENTS TO AQAP-2110 OR AQAP-2310

Edition B Version 1

AUGUST 2022

NORTH ATLANTIC TREATY ORGANIZATION

ALLIED QUALITY ASSURANCE PUBLICATION

Published by the
NATO STANDARDIZATION OFFICE (NSO)

INTENTIONALLY BLANK

NORTH ATLANTIC TREATY ORGANIZATION (NATO)

NATO STANDARDIZATION OFFICE (NSO)

NATO LETTER OF PROMULGATION

29 August 2002

1. The enclosed Allied Quality Assurance Publication AQAP-2210, Edition B, Version 1, NATO SUPPLEMENTARY SOFTWARE QUALITY ASSURANCE REQUIREMENTS TO AQAP-2110 OR AQAP-2310, which has been approved by the nations in the Life Cycle Management Group (AC/327), is promulgated herewith. The agreement of nations to use this publication is recorded in STANAG 4107.

2. AQAP-2210, Edition B, Version 1, is effective upon receipt and supersedes AQAP-2210, Edition A, Version 2, which shall be destroyed in accordance with the local procedure for the destruction of documents.

4. This NATO standardization document is issued by NATO. In case of reproduction, NATO is to be acknowledged. NATO does not charge any fee for its standardization documents at any stage, which are not intended to be sold. They can be retrieved from the NATO Standardization Document Database ((https://nso.nato.int/nso/) or through your national standardization authorities.

5. This publication shall be handled in accordance with C-M(2002)60.

Dimitrios SIGOULAKIS
Major General, GRC (A)
Director, NATO Standardization Office

INTENTIONALLY BLANK

RESERVED FOR NATIONAL LETTER OF PROMULGATION

INTENTIONALLY BLANK

RECORD OF RESERVATIONS

CHAPTER	RECORD OF RESERVATION BY NATIONS
Note: The reservations listed on this page include only those that were recorded at time of promulgation and may not be complete. Refer to the NATO Standardization Document Database for the complete list of existing reservations.	

INTENTIONALLY BLANK

RECORD OF SPECIFIC RESERVATIONS

[nation]	[detail of reservation]
Note: The reservations listed on this page include only those that were recorded at time of promulgation and may not be complete. Refer to the NATO Standardization Document Database for the complete list of existing reservations.	

INTENTIONALLY BLANK

TABLE OF CONTENTS

INTENTIONALLY BLANK

FOREWORD

The Acquirer's software quality assurance requirements stated in this document, are based on the experience that quality management of the entire software development process is the key to achieving software quality in complex and mission critical computer systems such as weapon systems, communication systems, and command and control systems. To ensure the quality of the software development process, such processes must be planned, controlled, and improved, with the aim of reducing, eliminating and, most importantly, preventing software quality deficiencies.

This document standardizes, to the greatest extent possible, the software quality management system requirements through harmonization of requirements from international software standards and other applicable documents and good practice.

The establishment of common requirements for use at all levels of the supply-chain by organizations around the world will result in improved quality, schedule, and cost performance by the reduction or elimination of organization unique requirements and wider application of good practice.

In accordance with international standardization, functional rather than organizational definitions for software quality management are used to avoid problems introduced by traditional quality concepts and their organizational boundaries. This publication, therefore, is not specifically addressed to software quality organizations but rather to the overall organizational structure and the different management levels involved in a software project.

This publication is designed for use in contracts and defines the requirements for the Software Quality Management Activities, as related to the Project, to be documented in a Project Software Quality Plan. The publication also requires the evaluation of the Software Quality Management Activities to ensure their effectiveness.

The application of this publication is not restricted to any particular type or form of software. This publication does not specify any particular software development model, nor does it stipulate which software development methods should be used. This publication allows flexibility in adapting the required documentation and procedures to the specific development and procurement processes of the project.

This publication supersedes AQAP-2210 Edition A Version 2 and is intended for use with AQAP-2110 or AQAP-2310, as a software-specific and project-oriented supplement.

INTENTIONALLY BLANK

CHAPTER 1 INTRODUCTION

1.1 General

This publication contains the NATO requirements for Software Quality.

1.2 Purpose

1. This publication contains requirements which, if applied appropriately, provide confidence in the Supplier's capability to deliver products that conform to Acquirer contract requirements.

2. This publication specifies the project-oriented requirements to manage the quality of the software development process. Both managerial and technical processes must be addressed in order to:

a. establish visibility of the software development process;
b. detect software quality problems as early as possible in the software life cycle;
c. provide quality control data for the timely implementation of effective corrective action;
d. confirm that quality is engineered in during the software development process;
e. provide assurance that the software produced conforms to contractual requirements;
f. ensure that appropriate software support is provided to activities at the system engineering level, if required by the contract;
g. ensure that the safety and security conditions of the project are addressed.

1.3 Applicability

1. This publication is primarily intended for use in a contract between two or more parties, in conjunction with AQAP-2110 or AQAP-2310.

2. When referenced in a contract, this publication shall apply to all the processes necessary for the Supplier to fulfil the contractual requirements.
In particular, this publication shall apply to:

a. all cases where software development is undertaken;
b. all cases where non-deliverable software is developed or employed under the contract;

c. all cases where software maintenance is part of the contract, in order to avoid uncontrolled, hidden development activities, which could have unforeseeable or detrimental consequences on the quality of the software product;
d. all cases where “Non-developmental/Off-The-Shelf” software (e.g.: Re-used software, Customer furnished software, COTS software, Government off-the Shelf (GOTS) software, Open-Source software) is to be delivered (see § 5.7.4);
e. all cases relating to the development of the software element of firmware.

3. Where identified by the Acquirer other appropriate standards can be used in conjunction with this publication to identify Management System process requirements.

4. This publication is intended for use with AQAP-2110 or AQAP-2310 as a software-specific and project-oriented supplement. Where there is any conflict between the requirements of AQAP-2110 (or AQAP-2310) and this publication for software, the requirements of this publication shall prevail.

5. If any inconsistency exists between the Contract requirements and this publication, the Contract requirements shall prevail.

6. This publication may also be used internally by a Supplier or a potential Supplier to cover the Software Quality aspects of the QMS (Quality Management System).

7. For competitive software acquisition this publication can also be used for the specification of requests for proposals and the evaluation of proposals. The provisions of this publication can also apply to Government Agencies performing software development or maintenance.

CHAPTER 2 COMPLIANCE WITH THIS PUBLICATION

2.1 Compliance

1. Compliance with this publication is defined as the fulfilment of the requirements in chapters 3, 4 and 5.

2. Whenever AS9115 rev A (or equivalent standard eg EN 9115:2018) uses the word “should”, this shall be read as “shall” and compliance by the Supplier is mandatory, unless otherwise determined by the Acquirer.

Note:

The replacement of the modal verb “should” with the modal verb “shall”, in the applicability of the AS9115 (see: § 4.1 and § 4.2), has added the need to introduce the “Tailoring” requirement in the AQAP-2210 (see § 4.3).

2.2 Notes and Guidance

In this publication “Notes” are not contractual requirements; they are for guidance or clarifying the associated requirement.

Note:

The paragraph numbers of the AS9115, mentioned (in brackets) at the end of some paragraph titles, are only for information purposes.

INTENTIONALLY BLANK

CHAPTER 3 COMPOSITION OF REQUIREMENTS IN AQAP-2210

3.1 Composition

1. The requirements in this publication are composed as follows:

   a. Chapter 4: “General Software QA Requirements”.
   b. Chapter 5: “NATO Specific Software QA Requirements”, that establishes additional NATO specific requirements for the Supplier.

2. This publication may only be used as a software supplement to AQAP-2110 or AQAP-2310, and therefore:

   a. If this publication is used as a Supplement to AQAP-2110 (which establishes the applicability of the requirements of ISO 9001), AS9115 will be applicable for the QMS (see § 4.1 of Chapter 4), and the NATO Specific QA Requirements for Software (see Chapter 5) will be related to AS 9115 and AQAP-2110.
   b. If this publication is used as a Supplement to AQAP-2310 (which establishes the applicability of the requirements of AS9100), the AS9115 will be applicable for the QMS (see § 4.2 of Chapter 4), and the NATO Specific QA Requirements for Software (see Chapter 5) will be related to AS9115 and AQAP-2310.

3. In case of contradiction, NATO specific requirements prevail over AS9115 requirements.

3.2 References

3.2.1 Normative References

1. AQAP-2110 NATO Quality Assurance Requirements for Design, Development and Production

2. ISO 9001:2015 Quality Management Systems – Requirements

3. AQAP-2310 NATO Quality Assurance Requirements for Aviation, Space and Defence Suppliers

4. AS9100 Rev. D Quality Management Systems – Requirements for Aviation, Space and Defence Organizations

5. AS9115 Rev A Quality Management Systems – Requirements for Aviation, Space and Defence Organizations – Deliverable Software (Supplement to AS9100)

3.2.2 Informative References

1. AQAP-2000 NATO Policy on an Integrated Systems Approach to Quality through the Life Cycle

2. ISO 9000:2015 Quality Management Systems – Fundamentals and Vocabulary

3. ISO/IEC/IEEE 90003:2018 Software engineering – Guidelines for the application of ISO 9001:2015 to computer software

4. ISO/IEC/IEEE 12207:2017 Systems and software engineering – Software life cycle processes

5. STANREC 4814 Systems and software engineering – Software life cycle processes

6. ISO/IEC/IEEE 15288:2015 Systems and software engineering – System life cycle processes

7. AAP-48 NATO System Life Cycle Processes

8. ISO/IEC 25010:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models (*confirmed in 2017*)

9. ISO/IEC 27001:2013 Information technology – Security techniques – Information security management systems – Requirements (*confirmed in 2019*)

10. AQAP-2105 NATO Requirements for Quality Plans

11. ISO 31000:2018 Risk Management – Guidelines

3.3 Definitions

The applicable definitions of ISO 9000 or AQAP-2110 or AQAP-2310 or AS9115 apply to the terminology used in this publication. Where definitions in ISO 9000 or AQAP-2110 or AQAP-2310 or AS9115 and this publication differ, the definitions in this publication shall apply.

1. Bug/Fault

Defect in a system or a representation of a system that if executed/activated can potentially result in an error. [Source: ISO/IEC/IEEE 15026-1:2019 Part 1]

2. Control

The activity to detect differences between an actual and a planned result/process, and to cause changes in a process or a product which reduce the detected differences to a defined level.

3. Critical items

Those items (e.g. functions, parts, software, characteristics, processes) having significant effect on the provision and use of the products and services; including safety, performance, form, fit, function, producibility, service life, etc.; that require specific actions to ensure they are adequately managed. Examples of critical items include safety critical items, fracture critical items, mission critical items, key characteristics, etc.

Critical items in software are those characteristics, requirements, or attributes that have been determined to be most important to achieve product realization (e.g., safety, maintainability, testability, usability, performance). For example, in the case of an aircraft's flight control system software, the response time could be elevated to a critical item to ensure overall performance characteristics are met; or if a project has customer specific testability requirements, cyclomatic complexity may become a critical item. [Source: AS9100 and AS9115]

4. Evaluation

A systematic determination of the extent to which an entity meets its specified criteria.

***Notes**:*

a. The term "entity" includes product, activity, process, organization or person;

b. Evaluation of the activity or process may occur in parallel with development, or may be deduced as the result of verification of the software product;

c. Evaluation of the activity or process can be performed by monitoring, auditing, process qualification or by establishing and documenting whether or not they conform to specified criteria.

5. Firmware

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

6. Integration

The process of combining software components, hardware components, or both into an overall system. [Source: ISO/IEC/IEEE 24765:2017]

> ***Note**:*
> *Significant Integration:*
> Integration which has been identified as having a major impact on performance and which requires specific action by the supplier to eliminate or to reduce the risk. This could be a single integration or the cumulative impact of multiple integrations.

7. Key characteristic

An attribute or feature whose variation has a significant effect on product fit, form, function, performance, service life, or producibility, that requires specific actions for the purpose of controlling variation.
Key characteristics in software are those measurable attributes where variability can be measured by the project and can, if left unchecked, adversely impact the project or product in areas (e.g., memory utilization, response time, functionality, reliability, usability, efficiency, maintainability, portability). [Source: AS9100 and AS9115]

8. Maintenance

Combination of all technical, administrative, and managerial actions during the life cycle of an item intended to retain it in, or restore it to, a state in which it can perform the required function. [Source: EN 13306: 2017]

9. Method

A set of rules for solving a problem.

10. Non-conformities (*classification*)

a. Major blocking: nonconformity that is major (see below) and prevents any significant use of the system.

b. *Major*: nonconformity that is likely to adversely affect performance; environment; safety; interchangeability; maintainability; reliability; service life or appearance of the product or when cost to the customer or delivery date agreed with the customer is likely to be affected. (*same definition in AQAP 2070 § A.3.2.1*).

c. *Minor:* all other departures from the specified technical requirements, which do not fall into the major or major blocking category, are considered minor *(same definition in AQAP 2070 § A.3.2.1 plus "*major blocking*").*

11. Non-deliverable Software

Software that is not required to be delivered under the contract but may be used in the development and/or support of other deliverable products.

12. Non-developmental Software

Deliverable software that is not developed under the contract, but is provided by the organization, customer, or a third party (e.g., reused software, customer furnished software, COTS software, Government off-the Shelf (GOTS) software, open-source software).

13. Regression testing/Non - regression testing

Regression testing (rarely non-regression testing) is re-running functional and non-functional tests to ensure that previously developed and tested software still performs after a change. If not, that would be called a regression. Changes that may require regression testing include fault fixes, software enhancements, configuration changes, and even substitution of electronic components.

14. Off-The-Shelf Software

Deliverable software that is already developed and usable as is, or with modification. Off-the-shelf software may be referred to as reusable software, Government furnished software, or commercially available software, depending on its source.

The Commercial-Off-The-Shelf (COTS) software is a commercially available application sold by vendors through public catalogue listings. COTS software is not intended to be customized or enhanced. Contract-negotiated software developed for a specific application is not COTS software. COTS software is a type of non-developmental software.

15. Patch

Modification made directly to an object program without reassembling or recompiling from the source program. [Source: ISO/IEC/IEEE 24765:2017]

16. Process

The interaction of personnel, equipment, material, and procedures aimed at providing a specified service or producing a specified product.

Each process is a defined set of one or more activities or tasks which can be accomplished in a finite period of time. Each process can be broken down into activities which are characterized by quantifiable inputs and outputs which can be measured, controlled, and improved.

17. Product safety

The state in which a product is able to perform to its designed or intended purpose without causing unacceptable risk of harm to persons or damage to property. [Source: AS9100]

18. Software Development Model

A simplified, abstract representation of the software development process (process behaviour and results) used for planning and control purposes.

19. Software Design and Development Process

The process by which user needs/requirements are translated into a software product.

20. Software Life Cycle

A framework containing the processes, activities and tasks involved in the development, operation, and maintenance of a software product, spanning the life of the system from the definition of its requirements to the termination of its use.

21. Software Quality Characteristics

A set of attributes of a software product by which its quality is described, verified, and validated. A software quality characteristic may be refined into multiple levels of sub-characteristics.

***Note**:*

According to the International Standard ISO/IEC 25010:2011, software quality may be evaluated using the following eight characteristics: Functional suitability, Performance efficiency, Compatibility, Usability, Reliability, Security, Maintainability, and Portability.

22. Software/Software Product

Computer programs, procedures, rules, associated documentation, and data, pertaining to the operation of a computer system.

23. Software Tool

A computer program used to help develop, analyse, evaluate, verify, validate, or maintain another computer program or its documentation.

24. Special requirements

Those requirements identified by the customer, or determined by the organization, which have high risks of not being met, thus requiring their inclusion in the operational risk management process. Factors used in the determination of special requirements include product or process complexity, past experience, and product or process maturity. Examples of special requirements include performance requirements imposed by the customer that are at the limit of the industry's capability, or requirements determined by the organization to be at the limit of its technical or process capabilities.

Examples of special requirements that may introduce high risk for software include: the introduction of a new compiler, new advanced modelling technique, qualification of tools, specific test equipment capabilities, introduction of a new type of interface, or specific customer technical requirements. These requirements are included in the risk management process. [Source: AS9100 and AS9115].

25. Validation

Confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use are fulfilled.

***Notes**:*

a. Validation is normally performed on the final product under defined operating conditions;

b. Multiple validations may be carried out if there are different intended uses.

26. Verification

The process of determining and obtaining objective evidence whether or not the products of a given phase of the software development process fulfil the requirements established during the previous phases.

***Notes**:*

a. Verification can be performed by reviewing, inspecting, testing, checking, auditing or otherwise establishing and documenting whether or not products conform to specified requirements.

b. A phase in this context does not imply a period of time in the development of a software product.

3.4 Acronyms

The following acronyms appear in this document:

COTS	Commercial Off-The-Shelf Software
EVV	Evaluation, Verification and Validation
GQAR	Government Quality Assurance Representative
MP	Maintenance Plan
PSQP	Project Software Quality Plan
QA	Quality Assurance
QMS	Quality Management System
SCI	Software Configuration Item
SCM	Software Configuration Management

INTENTIONALLY BLANK

CHAPTER 4 GENERAL SOFTWARE QA REQUIREMENTS

4.1 Supplementary Software Quality Assurance Requirements to AQAP-2110

1. If this publication is used as Supplementary Software Quality Assurance Requirements to AQAP-2110, the Quality Management System shall also be in accordance with this publication which includes the requirements of AS9115 as necessary to satisfy the contract requirements.

2. AS9115 (software standard related to the AS9100) will be applicable for the Software Quality Assurance Requirements, with the following clarifications.

3. In the context of AQAP-2210:

   a. References to AS9100 are replaced by ISO 9001.
   b. Paragraph 8.1.2 (Configuration Management) of AS9115 relates to paragraph 5.4.1.2 of AQAP-2110.
   c. Paragraph 8.1.4 (Prevention of counterfeit parts) of AS9115 relates to paragraph 5.4.6.2(2) of AQAP-2110.

4.2 Supplementary Software Quality Assurance Requirements to AQAP-2310

1. If this publication is used as a Supplementary Software Quality Assurance Requirement to AQAP-2310, the Quality Management System shall also be in accordance with this publication which includes the requirements of AS9115 as necessary to satisfy the contract requirements.

2. AS9115 (software standard related to the AS9100) will be applicable for the Software Quality Assurance Requirements.

4.3 Tailoring

1. The purpose of the Tailoring process is to adapt the requirements (and the related life cycle data) of the AQAP-2210 to satisfy the specific characteristics of the software project, and to ensure that the scope of the requirements (and life cycle data) selected is consistent with the risks associated with the software project.

2. The Tailoring process is the deletion (full or partial), downgrade, modification, or addition of processes (activities and tasks) of the AQAP-2210, in order to optimize the effectiveness and efficiency of the technical and managerial processes for the specific software project; tailoring is to be applied also to the life cycle data produced by each process (activity and task).

3. The Tailoring process shall be driven by project and product characteristics ("drivers" for tailoring); the drivers are either an implicit part of the project team's collective expertise or can be explicitly defined for a specific project or an application within a project; there are several types of drivers related to different aspects, such as dependability and safety, software development constraints, product quality and business objectives.

4. When tailoring the AQAP-2210, for use in a contract, the degree of tailoring shall be consistent with the characteristics of the product and the project, and the perceived risks.

5. Tailoring process can be performed at:

   a. the application (*SCI - Software Configuration Item*) level;
   b. the project (*Software System*) level.

6. Different tailoring intensity shall be applied, in general, as a function of the criticality and scope of development.

7. The initial approach to the tailoring shall be to understand the requested level of quality for the software project, with the characterization of the project, the identification of the needed processes, and the characterization of the product.

8. In order to apply the tailoring process, software shall be categorized in "criticality categories", based on the severity of the consequences of system failures, and in "development categories", based on the scope of the development activities that will be required to deliver the software.

4.3.1. Criticality Categories

1. The following four software "criticality categories" are defined, based on the severity of the consequences of system failures (including impact on availability, integrity, confidentiality, business, safety of persons, or damage to property), also in particular to determine the extent of integration testing or qualification to be applied:

- **A**
  Software that if not executed, or if not correctly executed, or whose anomalous behaviour, or whose incorrect integration into a system, can cause or contribute to a system failure resulting in:
  - **Catastrophic** consequences, in terms on safety effect or impact on mission success.
- **B**
  Software that if not executed, or if not correctly executed, or whose anomalous behaviour, or whose incorrect integration into a system, can cause or contribute to a system failure resulting in:

    - **Critical** or **Hazardous** consequences, in terms on safety effect or impact on mission success.
- **C**
  Software that if not executed, or if not correctly executed, or whose anomalous behaviour, or whose incorrect integration into a system, can cause or contribute to a system failure resulting in:
    - **Major** consequences, in terms on safety effect or impact on mission success.
- **D**
  Software that if not executed, or if not correctly executed, or whose anomalous behaviour, or whose incorrect integration into a system, can cause or contribute to a system failure resulting in:
    - **Minor** or **Negligible** consequences, with no safety effect or impact on mission success.

4.3.2. Development Categories

1. The following four software “development categories” are defined, based on the level of development, test, inspection, or control required to satisfy contract requirements:

- **1**
    - All bespoke software development or maintenance.
    - Re-used, open-source, or customer furnished software with significant integration.
- **2**
    - Re-used, open-source, or customer furnished software without significant integration.
    - Military or modified off-the-shelf software with significant integration.
    - Commercial off-the-shelf software with significant integration.
    - Bespoke non-deliverable software with significant integration.
- **3**
    - Military or modified off-the-shelf software without significant integration.
    - Commercial off-the-shelf software without significant integration.
    - Bespoke non-deliverable software without significant integration
- **4**
    - Commercial off-the-shelf software with minimal integration.

2. All tailoring decisions shall be agreed between the Customer and the Supplier and reflected in the contractual documents.

3. The Supplier, after the appropriate categorization of the software and the application of the tailoring process, shall prepare the PSQP with the final tailoring of AQAP-2210; all the tailoring decisions shall be documented together with the rationale for the decisions.

4.4. Tailorable/Non-tailorable Clauses

1. Only chapter 5 may be tailored. Within chapter 5, the following sub-clauses must not be tailored:

5.1 General
5.2 Project Software Quality Plan (PSQP)
5.3 Software Criticality Analysis
5.4 Software Quality Model
(*item # 2 can be tailored*)
5.5 Identification and Review of Software Requirements
(*item # 4 can be tailored*)
5.7.1 Organization roles, responsibilities, and authorities
5.7.2 Sub-supplier Management
5.7.3 Software Configuration Management (SCM)
5.7.5 Infrastructure

CHAPTER 5 NATO SPECIFIC SOFTWARE QA REQUIREMENTS

Note:

The paragraph numbers of AS9115, mentioned (in brackets) at the end of some paragraph titles, are only for information purposes.

5.1 General

1. To achieve visibility and control of the software development project the Supplier shall plan and implement effective software quality management activities.

2. The Supplier shall undertake a formal contract review to ensure all the contractual requirements are defined, and to determine the necessary management and technical processes which need to be planned and implemented.

3. Based on contract requirements, the rules and procedures of the organization's QMS, and the specific project requirements, the software quality management activities shall:

a. establish/identify, refine and allocate requirements to software products and configuration items (see: § 5.5);
b. establish and implement managerial and technical processes to design, develop, and build quality into the software (see: § 5.6 and § 5.7);
c. establish and implement procedures to verify and validate the quality of the software products and to evaluate processes and activities, including non-deliverable software, that impact the quality of the software products (see: § 5.8);
d. ensure that software specific risk has been included in the risk management plan. The Supplier shall identify, analyse, prioritize, and monitor the areas of the software project that involve software potential risk;
e. document and implement a secure software development methodology based on ISO/IEC 27001 or another recognized standard agreed with the Acquirer.

4. The project software quality management activities shall call upon existing standards and procedures in the organization's QMS. When this is not the case a justification shall be provided to the Acquirer and/or GQAR.

5. The project software quality management activities shall be documented in the Project Software Quality Plan (PSQP) (see § 5.2).

5.2 Project Software Quality Plan (PSQP) (8.1; 8.3.2)

1. The Supplier shall document the software quality management activities, as related to the Project, in a Project Software Quality Plan (see also AQAP-2105). The Project Software Quality Plan (PSQP) shall carry the signature of approval of those organizational elements having responsibilities identified in the PSQP and be placed under configuration control.

2. The Supplier shall submit an acceptable PSQP, which addresses the contractual requirements, to the Acquirer and/or GQAR in a mutually agreed timescale, but prior to the start of work which can be defined as a project or contract initiation meeting or as otherwise stated in the contract or purchase order. The PSQP shall be a clearly identified discrete document or part of another document that is prepared under the contract.

3. The PSQP shall document and maintain traceability of all the requirements from the project process by including a requirement and solution compliance matrix, justifying fulfilment of all the requirements of this publication (making reference where applicable).

4. The PSQP shall also include:

   a. analysis methods and criteria for determining software criticality (see § 5.3 "Software Criticality Analysis");
   b. quality models, derived from ISO/IEC 25010 or another appropriate standard (see § 5.4 "Software Quality Model"), for each software product, with the quality assurance and control rules for obtaining and assessing the sub-characteristics;
   c. the determination of the appropriate scope of operational planning and control required to support the software activities required by the contract.

Note:

Quality reports are to be positioned within the contract cycle (e.g., reviews, progress meetings, test phase, key point of the development cycle, etc.).

5. The PSQP shall be used by the Supplier as a current baseline to define the activities to monitor and control the quality of the software project. The PSQP shall be reviewed and updated at pre-defined milestones during the project as new definitions and development details become known.

6. The Acquirer and/or GQAR reserve the right to reject the PSQP and their revisions (see also AQAP-2105).

5.3 Software Criticality Analysis (4.3; 7.2; 8.1; 8.3.2; 8.3.4)

1. The Supplier shall provide a criticality analysis for each separate system software product and the associated criticality category, using the criticality analysis categories required or derived from the contract and being consistent with the criticality categories stated in the tailoring process (see § 4.3).

2. The criticality analysis shall be reviewed and updated at pre-defined milestones during the project, and as new definitions and development details become known.

5.4 Software Quality Model (8.1; 8.3)

1. The Supplier shall identify and provide a software quality model, for each software product, based on the ISO/IEC 25010 or another appropriate model. This quality model shall be adapted to each software product according to its criticality level, its preliminary risk analysis, and its context of development and use.

2. Maintainability aspects shall be considered for software products that are likely to evolve, such as software requiring regular maintenance.

3. The Supplier shall identify, with agreement from the Acquirer, the means of quality assurance and control for the software under test. These may include but are not limited to:

a. static measurements or metrics recorded during software development;
b. software requirement traceability tables linked to the software life-cycle items;
c. coding techniques and rules or standards;
d. inspection of items produced during development;
e. dynamic metrics based on tests;
f. non-regression assurance method (e.g., continuous integration);
g. checks of the technical risks (e.g., non-presence of memory leak, etc.);
h. source code reviews;
i. reinforced tests with integration of limit and off-limit cases;
j. updating and inspection of documentation produced;
k. appropriate means of Configuration Management.

Note:

Software products classed as "critical" will have more constraining thresholds for the metrics employed, traceability of the requirements for design, coding and tests, and test coverage imposed from the unit tests.

5.5 Identification and Review of Software Requirements (8.2)

1. The Supplier shall identify the software requirements and development constraints.

2. The software requirements specifications shall include a clear and precise definition of the design constraints and of the essential software quality characteristics.

3. The PSQP shall identify what standards or guides apply to the format and content of the software requirements specifications, either directly or by reference to procedures or documents.

4. The Supplier shall identify the functional, performance, interface, safety, and security specifications for each software product. The software specifications shall derive from the system or subsystem specifications.

5. Traceability matrices shall be provided by Supplier to show the derivation of software requirements from system or subsystem specifications.

6. If a software requirement review has not been performed as part of system development, it shall be an initial step in the software design and development process and be prescribed in the PSQP.

7. The review shall verify that software requirements are complete, consistent, unambiguous, traceable, feasible and can be validated. After the completion of the software requirements review, the software requirements specifications shall be formally approved by the Supplier's responsible authorities and shall be subject to Configuration Management.

5.6 Software Design and Development Process (8.3)

1. The Supplier shall apply a design and development model (lifecycle model) which breaks down the design into coding, verification, and validation activities. The rationale for the chosen model, and any risks it introduces, must be documented.

2. The Supplier shall produce an architectural and detailed design for each software product. The design shall present the modules or classes, the functions, the internal and external data, and the description of the inputs and outputs of each module.

3. The architectural and detailed design shall be updated for each corrective or progressive modification of the software.

4. The PSQP (see also AQAP-2105) shall cover Software Development addressing the requirements of AS9115 (§ 8.3.2) and the additions below:

a. an exhaustive list of the software products to which it applies;
b. the software lifecycle chosen and the justification for its use;
c. the schedule including initial planning for each software product with:

- development start and end dates with respect to the system development cycle;
- dates of integration in a subsystem and/or date of delivery and installation on site;
- reviews or milestones and key points identified by the Supplier for the development of each software product.

5. Where an item is not required, the PSQP must justify its omission.

6. For COTS products, where the Supplier does not have control over the design, the PSQP shall identify how the Supplier will assure that the product meets the acceptance criteria.

7. Any changes to development models, adopted during the project, must be recorded in the PSQP.

5.7 Management

5.7.1 Organization roles, responsibilities, and authorities (5.3; 7.1.2)

1. Personnel performing specific assigned tasks (outsourced labor or company employees) shall be qualified based on appropriate education, training, and/or experience as required; appropriate records shall be maintained.

2. A representative shall be appointed with the necessary authority to ensure all the requirements of this publication are met.

5.7.2 Sub-supplier Management (8.4)

Provision shall be made for Government Quality Assurance at the Sub-supplier's facilities when requested by the Acquirer and/or GQAR. When the Acquirer and/or GQAR determines that Acquirer and/or GQAR verification/validation/evaluation of the Sub-supplier's items/processes is necessary, the Supplier shall provide for this in the purchasing document. Copies of the purchasing document together with the relevant technical data shall be provided to the Acquirer and/or GQAR on request.

5.7.3 Software Configuration Management (SCM) (8.1.2; 8.5.1; 8.5.2; 8.5.6; 8.6)

1. The Supplier shall define and implement a SCM process to maintain integrity and traceability of the software product(s) during development. The SCM activities and procedures shall ensure that uncontrolled changes are prevented and shall provide

planned and released baselines as a reference and prerequisite for verification, tracing, and controlling software quality.

2. Specifically, the Supplier shall define and implement:

   a. procedures to identify, name and record the physical, functional, and quality characteristics of intermediate and final items to be controlled (e.g., documentation, executable code, source code, program listings, data bases, specifications, test cases, plans) and their structures at each project control point. Elements of the development and support environment (compilers, development tools, operating systems, test beds, licenses), i.e., “software tools”, shall also be part of the Software Configuration Item (SCI) structure;
   b. procedures to request, evaluate, approve/disapprove, and implement changes (error correction and enhancement) to baselined SCIs;
   c. procedures to record and report the status of project SCIs;
   d. audits and reviews for the determination to what extent the SCIs reflect the required physical, functional, and quality characteristics, and for establishing a baseline;
   e. procedures to control interfaces of project SCIs with items outside the direct scope of software development (system, hardware, human, support software);
   f. procedures to coordinate changes to externally developed software items and to incorporate those changes into the project.

3. Changes to the software requirement specifications shall be evaluated for cost, technical and schedule impact, and be communicated to all affected parties. Changes that will affect functional performance shall only be implemented with Acquirer approval.

4. The Supplier shall establish bidirectional traceability between the following elements:

   a. system requirements/software requirements/software component requirements;
   b. software requirements/validation tests/validation test sets/test results.

5. This traceability shall be retained in a matrix and/or in a dedicated tool.

Note:

This requirement is applicable to all software. In the case of COTS and Partially Modified Software, some traceability may not be possible or only partially.

6. The Supplier shall also identify the software tools, techniques and equipment which are necessary to implement SCM activities and allocate responsibilities and authorities for SCM activities to organizations and individuals within the project structure.

7. The Supplier shall provide, for each review, as from the internal validation phase, and for each delivery, a software release configuration report, listing at least the following items:

a. complete and unique identification of the software version;
b. identification of the delivery media;
c. list of applicable data: specification and interface documents, requests for upgrades;
d. procedure for generating executable code and the settings files if applicable (compilation instruction, links, etc.);
e. associated documentation produced during the development cycle;
f. complete identification of the configuration items for the delivered software product (including COTS); items that have changed from one version to the next shall be easily identifiable;
g. deviation or waiver requests with their status (ongoing, accepted);
h. list of faults/nonconformities corrected since the previous version;
i. list of ongoing faults/nonconformities with their classification (blocking, major, minor).

8. The use of patches is prohibited except for urgent corrections with agreement of the Acquirer. Configuration control of patches shall be prescribed in a specific procedure. Any patch shall only be temporary and must be included in a definitive version, with updated life-cycle data. The Supplier shall provide a patch configuration report including:

a. the impact study referenced for the technical incident (description of updates to be made, to the source lifecycle data, to obtain a software product identical to the patched software);
b. identification of the patch in the software version;
c. list of corrected faults/nonconformities;
d. installation procedure;
e. list of tests conducted for validating the patch, including non-regression tests;
f. exhaustive list of software configuration items contained in the patch.

5.7.4 Non-developmental/Off-The-Shelf Software (8.4)

1. If the Supplier employs deliverable Off-The-Shelf software, it shall ensure that:

a. its usability is unaffected by any existing data protection rights;
b. objective evidence exists, prior to its use, that the software will perform the required functions;
c. the software is placed under Configuration Management;
d. the software is documented in accordance with the requirements of the contract and this publication.

2. If deliverable Off-The-Shelf software is modified during the development process, such software shall then be treated as software under development and shall be subject to the requirements of this publication.

3. If the Supplier establishes that Off-The-Shelf software, supplied by the Acquirer, is not acceptable for use, it shall promptly report the reasons for its unacceptability to the Acquirer and negotiate the remedial actions to be taken.

4. The Supplier shall advise the Acquirer and/or GQAR when Off-The-Shelf software is to be incorporated into the software product.

5.7.5 Infrastructure (7.1.3)

The PSQP shall document the determination of the appropriate infrastructure to support the software activities required by the contract, either directly or by reference to procedures and documents.

5.7.6 Management Reviews (9.3)

5.7.6.1 General (9.3.1)

The top management review of the quality management system shall include the software specific aspects of the quality management system.

5.8 Evaluation, Verification and Validation (EVV)

5.8.1 General

1. The Supplier shall plan, define, and implement:

    a. a process for evaluation of software methods, techniques, procedures, tools and activities;
    b. a process for verification and validation of software items and software products;
    c. a process for the provision of follow-up action to ensure that necessary changes are made;
    d. a process to determine the required level of reverification in the case of error correction or change to the requirement/design.

2. The EVV process shall define:

    a. EVV activities and their sequence in relation to phases, milestones, and time schedule;
    b. the organizational roles, responsibilities, and authorities for the execution of EVV activities;

c. EVV objects (e.g., requirements/development documents, software products, development processes, methods, procedures, source code, object code);
d. the criteria to perform EVV activities;
e. specific EVV methods, standards, techniques, tools, and facilities;
f. the type of EVV methods to be used (e.g.: test, review, audit);
g. the EVV documentation to be produced (specific plans and procedures, EVV records and reports).

3. As an integral part of the EVV process the Supplier shall develop/select and implement quantitative and/or qualitative measures to evaluate/verify/validate the software quality characteristics specified in the requirements specifications or in the PSQP.

4. Quantitative/qualitative measures (metrics) shall also be applied to manage and control the software development process for the software product under contract. Such measures shall enable identification of the current level of performance, the taking of remedial action, and the establishment of improvement goals.

5. Personnel performing software quality evaluations, verifications and validations shall have the resources, responsibility, authority, and technical expertise. They shall also have an appropriate level of independence from the person(s) who developed the software product or performed the activity being evaluated/verified/validated, to permit objectivity and to cause the initiation of corrective action.

6. The Supplier shall be aware that Acquirer and/or GQAR EVV shall not constitute acceptance, nor shall it in any way replace EVV activities by the Supplier or otherwise relieve the Supplier of its contractual responsibilities.

5.8.2 Testing (8.3.4)

1. As an integral part of the EVV process the Supplier shall plan, define, and implement a test program. Consideration shall be given to:

a. software item, integration, system and acceptance testing;
b. test environment, tools, and test software;
c. qualification of the software test tools and test bed;
d. user documentation;
e. personnel required and associated training.

2. The Supplier shall undertake a review of test requirements and criteria for adequacy, feasibility, traceability and ambiguity. Test specifications shall be prepared which define test cases, required test data and expected results.

3. The Supplier shall define and implement measures to control test activities which include:

a. the establishment, documentation, and verification, as necessary, of the configuration of the software to be tested, together with any associated hardware;
b. the maintenance of test related documentation to allow test repeatability;
c. confirmation that tests are conducted in accordance with approved plans, specifications, and procedures;
d. provision for certification that test results are actual and valid;
e. provision for review and certification of test reports.

4. The Supplier shall report unusual difficulties found during test to the Acquirer and/or GQAR.

5.8.3 Reviews (8.2.3; 8.3.4)

1. Review procedures shall include provisions for:

a. describing the objectives;
b. identifying the functions, authorities, and responsibilities of personnel involved in the reviews;
c. recording review findings;
d. ensuring that actions resulting from reviews are monitored to ensure timely completion.

2. All software documentation generated under the contract shall be reviewed and approved for adequacy by authorized personnel prior to issue.

5.9 Maintenance

1. When, after initial delivery and installation, software maintenance is a specified requirement, the Supplier shall define and implement procedures for performing this activity. The Supplier shall provide, before the maintenance phase begins, a Maintenance Plan (MP) which addresses all requirements defined in this chapter. The MP may be a discrete document, or part of another plan that is prepared under the contract.

2. The procedures shall include provision for verifying and reporting that the maintenance carried out meets specified requirements.

3. Consideration shall be given to:

a. the work to be done;
b. the procedures to be employed;
c. the records and reports to be produced;
d. the responsibilities of the Supplier and its interface with the Acquirer and/or GQAR;

e. the Configuration Management activities, including the identification of the initial status of the product to be maintained;
f. the methods for dealing with the reporting, analysis and resolution of problems;
g. testing and acceptance of modifications.

AQAP-2210(B)(1)