STANDARDS RELATED DOCUMENT

AQAP-2210-SRD.1

NATO GUIDANCE ON THE USE OF AQAP-2210 NATO SUPPLEMENTARY SOFTWARE QUALITY ASSURANCE REQUIREMENTS TO AQAP-2110 OR AQAP-2310

Edition B Version 1
AUGUST 2022

NORTH ATLANTIC TREATY ORGANIZATION

Published by the
NATO STANDARDIZATION OFFICE (NSO)

INTENTIONALLY BLANK

NORTH ATLANTIC TREATY ORGANIZATION (NATO)

NATO STANDARDIZATION OFFICE (NSO)

NATO LETTER OF PROMULGATION

29 August 2022

1. The enclosed Standards Related Document, AQAP-2210-SRD.1 Edition B Version 1 NATO GUIDANCE ON THE USE OF AQAP-2210 NATO SUPPLEMENTARY SOFTWARE QUALITY ASSURANCE REQUIREMENTS TO AQAP-2110 OR AQAP-2310, which has been approved in conjunction with AQAP-2210 by the nations in the Life Cycle Management Group, is promulgated herewith.

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

3. 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.

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

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

INTENTIONALLY BLANK

TABLE OF CONTENTS

FOREWORD

This document has been prepared and issued to provide information and guidance on the application of AQAP-2210 Edition B Version 1 "NATO Supplementary Software Quality Assurance Requirements to AQAP-2110 or AQAP-2310".

It aims to contribute to commonality of interpretation of these requirements between Supplier and Acquirer. It is not intended as a procurement document. Its content has no legal or contractual status, nor does it supersede, add to, or cancel any of the AQAP-2210 requirements. Copies of this document may be made available to industry to facilitate the use and understanding of AQAP-2210.

Each paragraph (and sub-paragraph) of AQAP-2210 is listed in this document with its title, followed by the related guidance, or by the sentence "No guidance required".

Because of the multiplicity of conditions that can exist, this guidance should not be considered as all-encompassing, nor should it be considered as imposing specific means or methods for meeting contract requirements. Managers must be aware that other means or methods could be used to meet these requirements.

CHAPTER 1 INTRODUCTION

1.1 General

No guidance required.

1.2 Purpose

In addition to the requirements (a. through e.) strictly related to the software development process, this Publication also addresses the system-software relationship. The additional requirements (f. and g.) provide for the meaningful participation of software engineering in system engineering, and for addressing the system/software critical issues, like safety and security.

1.3 Applicability

No guidance required.

CHAPTER 2 COMPLIANCE WITH THIS PUBLICATION

2.1 Compliance

No guidance required.

2.2 Notes and Guidance

No guidance required.

CHAPTER 3 COMPOSITION OF REQUIREMENTS IN AQAP-2210

3.1 Composition

No guidance required.

3.2 References

3.2.1 Normative References

No guidance required.

3.2.2 Informative References

No guidance required.

3.3 Definitions

No guidance required.

3.4 Acronyms

No guidance required.

CHAPTER 4 GENERAL SOFTWARE QA REQUIREMENTS

4.1 Supplementary Software Quality Assurance Requirements to AQAP-2110

No guidance required.

4.2 Supplementary Software Quality Assurance Requirements to AQAP-2310

No guidance required.

4.3 Tailoring

1. The replacement of the modal verb “should” with the modal verb “shall”, in the applicability of AS9115, has added the need to introduce the “Tailoring” requirement in the AQAP-2210.

2. The goal of the Tailoring process should be the deletion (full or partial), downgrade, modification, or addition of processes (activities and tasks) of AQAP-2210, in order to optimize the effectiveness and efficiency of the technical and managerial processes for the specific software project.

3. In a similar way, tailoring should also be applied to the life cycle data produced by each process (activity/task). The produced information should be recorded into a document only if there is a need to have information transformed into a deliverable, recognizing that this transformation requires additional time and effort. On some projects, it may be desirable to combine deliverables, having then fewer deliverable documents to control, track and handle; to this extent it is important to also consider the timing of preparation, the intended audiences and the frequency of update of the deliverable documents.

4. This guidance does not address issues related to when (e.g. pre-contractual tailoring, tailoring by negotiation, post-contractual tailoring etc.) or by whom (Supplier, Acquirer or other) the tailoring of AQAP-2210 is performed.

5. The tailoring objectives should include quality improvement and consider benefits and efficiencies. To this extent all parties involved in a project should be involved in tailoring, and the decisions should be agreed between the Customer and the Supplier and reflected in the contractual documents; a team approach, involving all life cycle parties, could lead to significant benefits.

6. To perform the tailoring process, the software should be categorized, in order to determine the extent of test, inspection or control to be applied. This categorization should be based also on potential threats to availability, integrity and confidentiality, which may have an impact on business, safety of persons, or damage to property.

7. In approaching the tailoring, four major areas should be considered:

a. Both processes and data produced can and should be tailored.
b. It must be clear to those performing the tailoring processes, which of the requirements selected lend themselves to tailoring and which do not.
c. Tailoring is not an end in itself but there must be a logical reason behind the application of any tailoring process. The need for the adaptation of the requirements to a particular project is generally driven by considerations such as the level of software safety criticality, requirements for maintainability, specific development method such as Agile Methods or Formal Methods, specific project constraints (multinational, fixed price, delivery intervals, national security considerations, etc.), product characteristics, etc.
d. Tailoring is potentially dangerous if not properly controlled, and can lead to the selection of requirements that, in the short term, supports a fast and cheap project but ultimately can lead to a product that does not fulfil the contract. Therefore it is essential, throughout the tailoring process, to always respect the integrity and safety aspects of the final product.

TAILORING DRIVERS

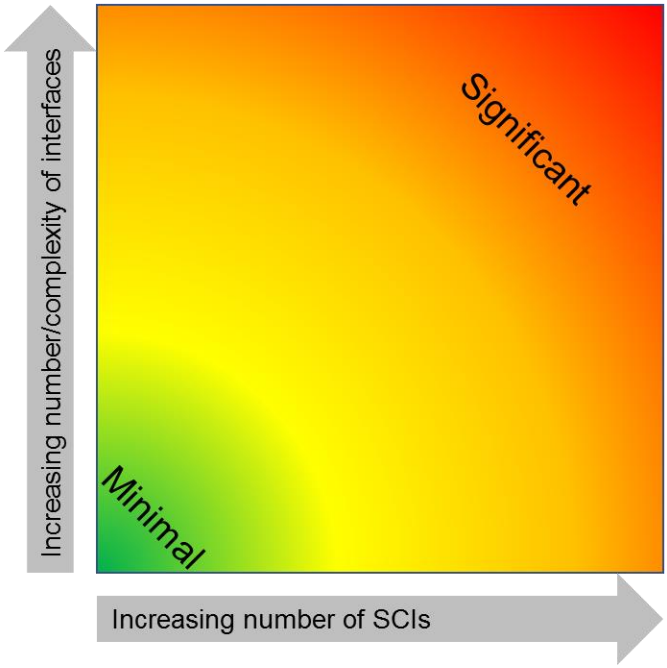
0. The tailoring process is driven by project and product characteristics (“drivers” for tailoring) and should always be done without affecting the quality, safety and integrity of the software product.

1. In making tailoring decisions, it is important to perform a risk analysis to balance short-term savings of cost and schedule against long-term risks. For example, tailoring out the products needed for software support can save time and money in the initial development, but may have severe negative effects on the long-term cost of supporting the software. In a similar manner; tailoring out software product evaluations, software testing and related activities, can save time and money in the short term but can result in reduced quality and expensive and time-consuming rework, once delivered and put into service.

2. There are several types of “drivers” for tailoring, related to different aspects, such as dependability and safety, software development constraints and methods, product quality and business objectives:

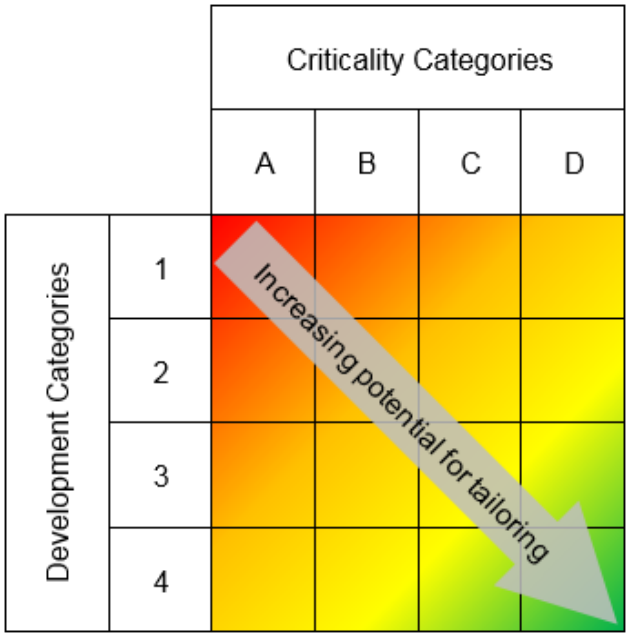
a. Tailoring for dependability and safety aspects should be based on the selection of requirements related to the verification, validation, and levels of proofs demanded by the criticality of the software.
b. Tailoring for software development constraints and methods should consider the special characteristics of the software being developed and of the development environment.
   The type of software development and the target system should also be considered; specific requirements for verification, review, and inspection should be imposed, for example, when full validation on the target device is not feasible or where performance goals are difficult to achieve.

Some factors influencing integration significance



c. Tailoring for product quality and business objectives should be done by selecting requirements for quality of the product, based on the quality objectives for the product specified by the Customer.

3. The diagram below shows the increasing potential for tailoring, as the level of criticality and the scope of development reduce.



4. There are also “drivers” related to technical, operational, and management aspects that influence the tailoring process:

a) Technical drivers:
- novelty of the domain of application;
- complexity of the software and the system;
- criticality level;
- size of the software;
- reusability required of the software being developed;
- interface to system development projects;
- degree of use of COTS or existing software;
- maturity of the COTS and completeness or stability of the user requirements.

b) Operational drivers:
- type of application;
- number of potential users of the software;
- criticality of the software as measured by the consequences of its failure;
- expected lifetime of the software;
- number of sites where the software is used;
- operation, maintenance, migration, and retirement constraints.

c) Management drivers:
- amount of time and effort required to develop the software;
- budget requirements for implementing and operating the software;
- accepted risk level for the project;
- type of life cycle;
- schedule requirements for delivering the software;
- number of people required to develop, operate and maintain the software;
- complexity of the organization;
- experience of the Supplier;
- financial resource.

5. The table below shows (only as a simple illustration the interrelationship between drivers and processes) a non-exhaustive correlation between the “drivers/characteristics” and the “processes” mainly affected.

	Driver/ Characteristic	Acquisition	Supply	Development	Operation	Maintenance	Documentation	Configuration Management	Quality Assurance	Verification	Validation	Joint Review	Audit	Problem Resolution	Management	Infrastructure	Training
Product	system complexity			X	X	X	X	X	X		X						
	software complexity			X	X	X	X	X	X		X						
	software size			X					X						X		
	hardware/software resources constraints			X					X	X							
	safety			X	X		X	X	X	X	X	X		X			
	security			X	X		X	X									
	reliability			X	X			X	X	X	X	X					
	maintainability			X		X	X	X	X								
	interoperability			X	X		X	X			X						
	availability				X						X						
	usability			X	X		X				X						
	level of reuse			X					X	X							
	use of new technologies			X					X							X	X
Project	project schedule constraints	X	X									X	X		X		
	project resources constraints	X	X									X			X		
	contractual complexity	X	X	X				X	X			X			X		

TAILORING STEPS

1. The main steps to tailor the AQAP-2210 should be:

a) Identify the project environment and characteristics
Characteristics of the project environment, and the circumstances that are going to influence the tailoring should be identified and recorded.

This includes but may not be limited to:

- stability of and variety of operational requirements;
- risks (commercial or performance) to the concern of interested parties;
- novelty, size, complexity and criticality;
- hardware resourcing;
- use of a specific language;
- use of an independent verification and validation agent;
- integrity issues such as safety, security, privacy, usability, availability;
- emerging technology opportunities;
- profile of budget and organizational resources available;
- starting date and duration of utilization;
- number of personnel and parties involved;
- roles, responsibilities, accountabilities, and authorities in the overall life cycle;
- pertinent laws and regulations;
- need to conform to other standards.

b) Solicit inputs
Inputs from the organizations that are to be affected by the tailoring decisions should be solicited.

Each organization should have an opportunity to state its needs and to take advantage of the combined expertise of all participants; users and support personnel should be also involved in tailoring.

This includes but may not be limited to:

- the system stakeholders;
- the interested parties to the agreement made by the organization;
- the contributing organizational functions.

c) Select development strategy
The development strategy most relevant and applicable (such as waterfall, evolutionary, incremental, or spiral) should be determined. Each such strategy prescribes certain processes (activities and tasks) that may be performed sequentially, repeated and combined.

d) Select stages and processes

The relevant stages of the life cycle model should be identified, as well as their entry and exit criteria and their relationship (serial, parallel, wholly or partly combined).

The life cycle processes (activities and tasks) and the related life cycle data that require tailoring should be selected.

e) Document the tailoring decisions and rationale

All tailoring decisions should be documented together with rationale for decisions.

The Supplier must rationalize the basic principle of its tailoring and the output of the tailoring should be documented in the PSQP.

A suggested format for documenting the tailoring decisions is at **Annex A**.

4.4 Non-tailorable clauses

No guidance required.

CHAPTER 5 NATO SPECIFIC SOFTWARE QA REQUIREMENTS

5.1 General

1. AQAP-2210 normally presumes the existence of a documented organization's Quality Management System (QMS), related also to the Software aspects, that includes not only the technical processes of the software design and development, but also the managerial processes.

2. The QMS should address the range of software that the Supplier produces. Different methods, procedures and tools may be called for dependent on the type of application, size of project, number of people involved etc.

3. A periodic, systematic and documented evaluation of the status and adequacy of the QMS should be conducted by or on behalf of, top management to ensure that their objectives are reached and to reveal non-conformances or irregularities in the system elements that require improvement.

4. For the QMS to be effective it should support the requirements of AQAP-2210 and any additional requirements imposed by the contract.

5. The Project Software Quality Management Activities should comprise the planning and implementation activities necessary for the successful execution of the project. The project activities mentioned in the paragraph 5.1 (items 3a, 3b, 3c, 3d) are elaborated in paragraphs 5.5 through 5.8. Guidance on these activities is given on each paragraph.

6. It is important to take note that the item 3c refers also to non-deliverable software (which may be employed in the development of deliverable software, like emulators, tools for autogenerated code and/or autogenerated testing, test harnesses and driver programs, stub routines, etc.); it is essential that all such software is placed under configuration management, since it directly affects the integrity of the deliverable software.

7. The depth of Project Software Quality Management Activities will be influenced by the contractual requirements and constraints; like complexity, criticality, size, Acquirer involvement etc. Therefore, as a prerequisite to the planning of the activities, the Supplier should undertake a formal contract review to ensure all requirements and constraints are clearly defined and understood.

8. Evaluation of the activities by the Acquirer should initially make use of objective evidence of the Supplier's own reviews. Where no objective evidence exists that such reviews have been conducted, it should be regarded as a serious quality system shortcoming and consequential risk.

5.2 Project Software Quality Plan (PSQP)

1. The PSQP and its contents should be recognized by Acquirer and Supplier as an indication of the understanding, commitment, and compliance with the quality requirements of the contract.

2. Suppliers should begin to plan their quality related activities at the earliest possible phase of the contract.

3. Quality reports may be in the form of EVV (see § 5.8) reports, test results, corrective action reports, etc. They can be the formal results of both main Supplier and Sub-supplier activities.

4. The PSQP should address "contract-specific" quality activities and should not be a reiteration of the QMS requirements, as detailed in the Supplier's Quality Manual/Documentation; however, reference to these requirements in the PSQP may be necessary.

5. Quality planning provides the means for tailoring the application of the QMS to a specific project, product or contract.

6. A PSQP may be required in response to an “Invitation-to-Tender / Request-for Proposal”, or under the contract, and should be prepared as a precursor to the software design and development process.

7. A possible layout for the PSQP may be found in AQAP-2105.

8. Further guidance may be found in ISO/IEC/IEEE 90003:2018 (§ 8.1.1).

5.3 Software Criticality Analysis

No guidance required.

5.4 Software Quality Model

No guidance required.

5.5 Identification and Review of Software Requirements

1. The software requirements may be derived from an implied need (but not necessarily specified) by the Acquirer. Often the Supplier does not fully understand the Acquirer's problem and field of application. Both contractual parties may work together to come to a formal contractual agreement on what the completed software must do.

2. The key to achieving an effective software design and development should be, for both the Supplier and the Acquirer, the achievement of a common understanding of the requirements. Therefore, the Supplier should ensure that the software requirements are described in such

a way that their interpretation is not in doubt. Any omissions, misunderstandings or inconsistencies in the requirements should be addressed, as early as possible, in the software design and development process, when they are easier to correct. The Supplier should also be satisfied that each requirement is defined in such a manner that its achievement can be ultimately subjected to validation by a prescribed method. If this is in doubt, the matter should be brought to the attention of the Acquirer.

3. Often the software requirements are derived from higher level (system or subsystem) requirements; therefore, it is important to ensure that all applicable higher-level requirements have been correctly translated into software requirements and no new requirements have been introduced.

4. "Development constraints" are restrictions on the design and development process, which shift greater design responsibility to the Organization setting the restrictions and need to be separated from the software quality characteristics. Examples are:

- Design standards and conventions,
- Languages,
- Computer hardware, and
- Acquirer supplied software.

5. Definitions of software quality characteristics can be found in ISO/IEC 25010:2011 (*confirmed in 2017*). These include:

- Functional suitability,
- Performance efficiency,
- Compatibility,
- Usability,
- Reliability,
- Security,
- Maintainability, and
- Portability.

6. Further guidance may be found in ISO/IEC/IEEE 90003:2018 (§ 8.2.2).

5.6 Software Design and Development Process

1. The software design and development process has a significant impact on the quality of the software product.

2. Software design and development models are simplified, abstract representations of a systematic approach to the software design and development process and, together with methods and tools, are important quality management elements, and should satisfy the following quality related criteria:

- reduces the complexity of the development process to ensure visibility and control;
- describes software and system integration;
- describes the software system architecture;

- makes use of recognized software engineering practices;
- utilizes data feedback from previous designs;
- describes the activities and their expected results clearly;
- identifies tasks which are critical to quality and project success;
- defines and chronologically assigns control points at which the correct course of the process and the correct transfer of results can be verified;
- describes how unplanned activities will be controlled;
- provides unambiguous start and end criteria for all processes;
- provides clear identification and allocation of all quality functions within the project specific organizational structures;
- uses proven and qualified constructive and analytical quality measures;
- provides quality data for the effective management of the development process;
- relates planning, monitoring and release activities to software engineering activities;
- reduces the risk by using computer resources to free people involved in the software development process from error prone, repetitive activities.

3. There are several types of model (e.g. Waterfall, Evolutionary, Incremental, Spiral, etc.) and AQAP-2210 gives the Supplier freedom in the choice of the model. The selection, definition and application of a specific model depends on the complexity, criticality, and type of software to be developed. Whatever model is selected, it may be tailored to meet the specific contract requirements. Where possible account should be taken of International or National standards defining these models.

4. The model should clearly describe all the primary (technical) processes (e.g., design, coding, validation, etc.) together with the supporting (project) processes (e.g., project management, configuration management, etc.) and the organizational (organizational project-enabling) processes (e.g., human resources, infrastructure, etc.) undertaken throughout the software lifecycle. The description of the processes should not only include the identification of the activities and tasks but also the roles (architect, tester, etc.), the inputs, the outputs, the start and end criteria, the control points, the related measurement (when applicable) and all the technical and managerial aspects. This is in order to reduce the complexity of the software design and development process, thus giving improved visibility, integrity and control of the software product itself.

5. For the software design and development activities, the Supplier should employ recognized software engineering methods, tools, resources, and procedures. The Supplier should also identify and standardize specific conventions for any graphical or formal linguistic notations. The methods, tools, standards and procedures used should support the software lifecycle to:

- express software requirements including quality characteristics;
- translate the Acquirer/User oriented software quality requirements into software engineering-oriented characteristics and allocate these to the appropriate level of design;
- ensure traceability at all design and implementation levels;
- minimize errors;
- support EVV (Evaluation Verification and Validation) activities during software design and development.

6. The methods, tools and procedures used should be evaluated and documented and should support the recognized principles and concepts of software engineering that influence software quality.

7. Software tools may be related to the specific methods or techniques identified or provide support to other aspects of the software lifecycle. Some tools may be phase-independent, for example those associated with configuration management or quality assurance activities.

8. Software tools should be validated to confirm their performance and integrity by a defined method, and may entail one or more of the following:

- certification provided by a recognized body that the tool has been subject to specified tests or validation processes;
- establishment, with the tool supplier, that the tool meets required criteria through the Supplier quality system and evidence of appropriate tests;
- identification of appropriate tests to be applied to the tool and any upgrades;
- monitored usage of the tool during support to the development of the software product;
- feedback from a user group.

9. Design and development models are a basis for detailed planning of project software quality management activities, including time and budget aspects and support continuing improvement of the software design and development process.

10. Further guidance may be found in ISO/IEC/IEEE 90003:2018 (§ 8.3.2.1 and § 8.3.2.2).

5.7 Management

5.7.1 Organization roles, responsibilities, and authorities

1. It is important to define the interrelationship of organizational elements and groups since activities may overlap and be executed iteratively. It is also important that the organizational structure indicates the co-operation and consultation between elements or groups and indicates the point(s) of contact with the Acquirer.

2. The degree of independence required for personnel performing evaluations, verifications, and validations, may depend upon the circumstances of the particular contract and/or Supplier concerned. In most instances, suitably independent personnel may be found amongst the peers of those who developed the software product or performed the activity being subjected to evaluation, verification or validation. Sometimes it may be necessary to seek such personnel within other areas or organizations, internal or external to that of the Supplier. Where special independence requirements pertain, such as for safety critical software, these should be defined in the contract.

3. A determination of the necessary independence of the verification effort should be required based on the potential of an undetected error in a system or software for causing:

- death or personal injury;
- mission failure;
- catastrophic equipment loss or damage;
- the maturity of, and risks associated with, the software technology to be used;
- financial loss.

4. Consideration should be given to the provision of training of personnel (both Acquirer and Supplier); personnel performing specific assigned tasks should be qualified, based on appropriate education, training and/or experience as required:

- design methods;
- specific programming languages;
- tools, techniques;
- computer platforms and target environment.

5. Further guidance may be found in ISO/IEC/IEEE 90003:2018 (§ 5.3 and § 7.1.2).

5.7.2 Sub-supplier Management

1. The main Supplier should be responsible for ensuring that sub-contracted products and services comply with the requirements and conditions of the main contract, even if the entire software package is sub-contracted.

2. The Supplier should select Sub-suppliers, using an appropriate procedure, based on their ability to meet sub-contract requirements, including quality. The Sub-suppliers previously demonstrated performance should also be considered.

3. The Sub-supplier's PSQP should be related to the main Supplier's PSQP. This relationship of plans is necessary for configuration management and specifically to coordinate changes to configuration items.

4. Further guidance may be found in ISO/IEC/IEEE 90003:2018 (§ 8.4.1.1 and § 8.4.1.2).

5.7.3 Software Configuration Management (SCM)

1. In software development and/or maintenance a strong relationship exists between SCM and software quality assurance. Without a disciplined SCM process one of the means for quality assurance is missing.

2. Configuration management is a discipline for identifying, controlling, tracking, and auditing the versions of each SCI (Software Configuration Item). SCM should be applied in a cost-effective manner, in terms of organization, methods, tools and procedures, whilst ensuring the necessary integrity and traceability of the software product.

3. Temporary changes to delivered software, sometimes known as “patches”, should be strictly controlled. Where such changes are introduced into software they should be carried out in accordance with defined procedures. In any event, follow-up action should confirm the validity of the change and where appropriate formally introduce it under normal configuration management procedures.

4. Non-conforming software should be clearly identified as such and segregated from conforming software.

5. Once "conforming" software is released and made available for use (e.g. to test areas or placed in the software library) its status should be clearly indicated and made known.

6. Upon becoming “non-conforming” software (e.g. after failing a test or a confirmed customer fault report) it should be segregated, clearly indicating its non-conforming status and taking appropriate action to control access to the software.

7. Further guidance may be found in ISO/IEC/IEEE 90003:2018 (§ 8.5.2).

5.7.4 Non-developmental/Off-The-Shelf Software

1. The evaluation and validation of the ability of the Off-The-Shelf software to perform the required functions, should include such considerations as Intellectual Property Rights, licensing arrangement, and Configuration Management controls.

2. Off-The-Shelf software should be placed under Configuration Management as it affects the integrity of the developed software whether it is a component of the software under development or a tool to assist the development of such software.

3. The Supplier should be able to provide objective evidence (e.g. validation reports, configuration reports, etc.) that the use of the Off-The-Shelf software has been evaluated and is under control.

4. Documentation requirements for the Off-The-Shelf software should include functional and interface specifications.

5. Off-The-Shelf software includes "Government furnished software". Government furnished software places constraints on the Supplier in terms of development freedom and responsibility.

6. Further guidance may be found in ISO/IEC/IEEE 90003:2018 (§ 8.4).

5.7.5 Infrastructure

7. The Supplier should describe the storage, storage security, environment, access to and release from storage in a procedure that also indicates how these activities are controlled.

8. Any media on which software is stored should be handled in such a way that the integrity and confidentiality of the stored information is assured.

9. It is therefore necessary that activities likely to influence the quality are recognized and steps taken to avoid degradation of the material or the information.

10. Adequate antivirus and firewall protection should be provisioned.

11. Further guidance may be found in ISO/IEC/IEEE 90003:2018 (§ 7.1.3).

5.7.6 Management Reviews

No guidance required.

5.8 Evaluation, Verification and Validation (EVV)

5.8.1 General

1. Although EVV is an integral part of the management and technical process (§ 5.5, 5.6, 5.7 and 5.9), due to its importance in Quality Management the EVV process is addressed in this discrete paragraph.

2. Due to the interrelationship of Evaluation, Verification and Validation, these activities should be planned as a whole. The allocation of resources and time, and the selection of methods and techniques, should be done in such a way that the entire EVV process is optimized.

3. The correct execution of EVV tasks has a considerable impact on the quality of the final product. This process requires, in general, the use of a considerable amount of resources, so that it should be carefully planned in terms of availability of qualified personnel, schedule, cost and test environment.

4. The level of EVV should be tailored to the level of complexity and/or criticality of the software and to the requirements of the contract and should involve optimum use of existing techniques and standards available.

5. Verification criteria for software items, consistent with the software design, should allow a software integration strategy where:

- software items are developed that ensure compliance with the software requirements allocated to the items;
- software items are verified using the defined criteria;
- software items defined by the integration strategy are produced;
- results of integration testing are recorded;
- consistency and traceability are established between software design and software items;
- a regression strategy is developed and applied for re-verifying software items when a change in software items (including associated requirements, design and code) occur.

6. This paragraph is also related, as a by-product, to the evaluation and improvement of the Software aspects of the QMS, e.g., it monitors the application of the established procedures, and measures the correctness and efficiency of these procedures. This evaluation process should be based on data provided by project groups and is a contract-independent activity.

7. Further guidance may be found in ISO/IEC/IEEE 90003:2018 (§ 8.3.4.1, § 8.3.4.2 and § 8.3.4.3).

5.8.2 Testing

1. In general, tests are much more effective the earlier they are addressed/conducted in the software development process.

2. Planning for testing and the specification of tests should therefore take place as early as possible. During test planning, if required by the contract, consideration should be given to the involvement of Acquirer personnel in test activities.

3. Further guidance may be found in ISO/IEC/IEEE 90003:2018 (§ 8.3.4.4).

5.8.3 Reviews

1. Software-related review activities may be known under various headings, including design reviews, peer reviews, walk-throughs, inspections, document reviews, desk checks etc.

2. Experience has shown that significant software errors are introduced during the early phases of the software development process. Emphasis should therefore be placed on design reviews at these stages to promote the early detection and resolution of errors.

3. Further guidance may be found in ISO/IEC/IEEE 90003: 2018 (§ 8.3.4.1).

5.9 Maintenance

1. For the software maintenance activities, the Supplier should also consider:

- the nature, use and intended lifetime of the software product;
- customer requirements and feedback;
- collection/analysis of in-service data (performance, reliability, lessons learned, …);
- control, updating and provision of technical documentation relating to the use of the software product;
- controls required for work undertaken external to the organization;
- product/customer support (queries, training, warranties, obsolescence, …).

2. When problems are detected after delivery, the Supplier should take appropriate action including investigation and reporting.

3. To facilitate software product maintenance, the availability of longer-term support for software tools is an important aspect that should not be ignored.

4. Further guidance may be found in ISO/IEC/IEEE 90003:2018 (§ 8.5.1.7).

ANNEX A

Template for AQAP-2210 Tailoring Justification

Justification: The reason why the clause/sub-clause is not applicable to the contract and can be tailored out.

Risk: The risk event(s) and consequence on the achievement of objectives that may occur as a result of the clause/sub-clause being tailored out.

Mitigation: An activity or measure (if required) that is expected to reduce the impact of the risk event should it occur.

Control: An activity or measure (if required) that is expected to reduce the likelihood of the risk event occurring.

Tailoring of AQAP-2210 Ed.B Ver.1 Clauses

Clause	Title	Sub Clause	Tailored Out	Justification	Risk	Mitigation	Control
5.1	General	All	Cannot be tailored out				
5.2	Project Software Quality Plan (PSQP)	All	Cannot be tailored out				
5.3	Software Criticality Analysis	All	Cannot be tailored out				
5.4	Software Quality Model	1, 3	Cannot be tailored out				
	Software Quality Model	2					
5.5	Identification and Review of Software Requirements	1, 2, 3, 5, 6, 7	Cannot be tailored out				
	Identification and Review of Software Requirements	4					
5.6		1					

Clause	Title	Sub Clause	Tailored Out	Justification	Risk	Mitigation	Control
	Software Design and Development Process	2					
		3					
		4					
		5					
		6					
		7					
5.7	MANAGEMENT						
5.7.1	Organization roles, responsibilities, and authorities	All	Cannot be tailored out				
5.7.2	Sub-supplier Management	All	Cannot be tailored out				
5.7.3	Software Configuration Management (SCM)	All	Cannot be tailored out				
5.7.4	Non-developmental / Off-The-Shelf Software	1					
		2					
		3					
		4					
5.7.5	Infrastructure	1	Cannot be tailored out				
5.7.6	Management Review						
5.7.6.1	General	1					
5.8	EVALUATION, VERIFICATION AND VALIDATION (EVV)						
5.8.1	General	1					
		2					
		3					
		4					
		5					

Clause	Title	Sub Clause	Tailored Out	Justification	Risk	Mitigation	Control
		6					
5.8.2	Testing	1					
		2					
		3					
		4					
5.8.3	Reviews	1					
		2					
5.9	MAINTENANCE	1					
		2					
		3					

Tailoring of AS9115 Clauses

Clause	Title	Tailored Out	Justification	Risk	Mitigation	Control
4.3	Determining the scope of the quality management system					
4.4	Quality management system and its processes					
6.1	Actions to address risks and opportunities					
7.1.3	Infrastructure					
7.1.4	Environment for the operation of processes					
7.1.5.1	General					

Clause	Title	Tailored Out	Justification	Risk	Mitigation	Control
7.2	Competence					
7.3	Awareness					
7.5.3	Control of documented information					
8.1	Operational planning and control					
8.1.1	Operational risk management					
8.1.1.1	Product integrity					
8.1.2	Configuration management					
8.1.3	Product safety					
8.1.4	Prevention of counterfeit parts					
8.2.2	Determining the requirements for products and services					
8.2.3	Review of the requirements for products and services					
8.3.2	Design and development planning					
8.3.3	Design and development inputs					

Clause	Title	Tailored Out	Justification	Risk	Mitigation	Control
8.3.4	Design and development controls					
8.3.5	Design and development outputs					
8.3.6	Design and development changes					
8.4.1	General					
8.4.2	Type and extent of control					
8.4.3	Information for external providers					
8.5.1	Control of production and service provision					
8.5.1.1	Control of equipment, tools, and software programs					
8.5.1.3	Production process verification					
8.5.2	Identification and traceability					
8.5.4	Preservation					
8.5.6	Control of changes					
8.6	Release of products and services					

Clause	Title	Tailored Out	Justification	Risk	Mitigation	Control
8.7	Control of nonconforming outputs					
9.1.3	Analysis and evaluation					
9.2	Internal audit					
9.3.2	Management review inputs					

AQAP-2210-SRD.1(B)(1)