Annex A

(informative)

Software requirements and architecture review (SAR)

A.1 General

Some department of defense (DoD) domains may require a SAR due to the nature and complexity of the system(s) being developed.

A.2 Annex A purpose

Annex A provides for the SAR the corresponding content contained in Clause 5, Clause 6, and Clause 7 of this standard for the reviews covered in those clauses.

A.3 Annex A tailoring

If a specific program requires a SAR, Annex A content changes status as defined in A.4.

A.4 Application of Annex A content

For programs that require a SAR, the content of A.5, A.6, and A.7 assume the status and are applied as listed in Table A.1.

Table A.1—Annex A content status and application

Annex A subclause	Content status	Application
A.5	Becomes normative	Is a normative addition to Clause 5 of this standard.
A.6	Becomes normative when tailored	Is a normative addition to Clause 6 of this standard as tailored for the program.
A.7	Remains informative	Is added as application guidance to Clause 7 of this standard.

A.5 Requirements for a SAR

A.5.1 SAR purpose

The SAR shall be conducted to confirm that a selected set of system software elements' requirements, logical architecture, test planning, development processes, and current state of development form a satisfactory basis for proceeding to the preliminary software design cycle.

A.5.2 SAR description

The SAR shall confirm that

a) The software requirements and architectural design are adequate for meeting the higher-level requirements allocated to software.

b) The software requirements and architectural design are sufficiently mature to proceed with dependent software and system development activities.

c) The software processes are sufficiently defined, mature, and effective for developing the software needed to meet system requirements and operational needs, and are suitable for the program scope and complexity.

d) The software test plans are sufficiently robust to ensure thorough testing of the software products to demonstrate that the software requirements are verified in the target environment.

e) The software development and test environments are established and have adequate capability and capacity to meet the software development and test requirements and schedules.

f) The software requirements specifications (SRS), IRS, software test plan(s) (STP), software architecture description (SAD), and software operational concept document form a satisfactory basis for proceeding to preliminary software design.

g) The software development risk is manageable.

h) The software development costs and schedules are consistent with program costs and schedules.

A.5.3 SAR timing

A.5.3.1 SAR event-based scheduling

The SAR shall be held after the system functional review (SFR), when the selected set of software elements' requirements and logical architecture have been sufficiently defined to evaluate the supplier's responsiveness to and interpretation of the system, subsystem, or prime item-level requirements.

A.5.3.2 SAR scheduling constraints for software life-cycle model

Due to differences in the state of the various software development products as a function of the chosen software life-cycle model, the following additional timing constraints apply:

a) When the software element(s) under review are being developed using a waterfall life-cycle model, the SAR shall be completed prior to conducting the system preliminary design review (PDR).

b) When the software element(s) under review are being developed using an incremental or evolutionary life-cycle model, the SAR shall be conducted at the conclusion of the software architectural design for each incremental or evolutionary build.

A.5.4 SAR entry criteria

The SAR shall be conducted only after the following events have been successfully completed:

a) The acceptability criteria for each of the SAR technical review products have been established for the specific program by tailoring the contents of Table A.2.

b) All preparatory actions in Table A.3 as tailored for the specific program have been successfully accomplished to support conducting the technical review.

c) All documentation in support of the SAR has been completed to the degree to satisfactorily support the technical review.

A.5.5 SAR content

A.5.5.1 Products to be reviewed at SAR

The following work products at a minimum shall be reviewed by the SAR team. Other products may be added as necessary during tailoring of Table A.2 for the specific program.

a) Software technical documentation

b) Technical plans

c) Program execution and process control

d) Risk assessment

e) Program cost and schedule estimates

A.5.5.2 Conduct of the SAR

SAR participants shall assess the SAR work products and judge the products' acceptability according to the applicable criteria in Table A.2 as tailored for the specific program.

A.5.5.3 SAR outputs

a) Key SAR outputs shall include the following:

   1) Approved software requirements and logical architecture baselines that are under configuration control

   2) Updated software cost and schedule estimates consistent with the level of completeness defined in the software development plan (SDP), based on the selected life-cycle model

   3) An updated (if necessary) risk and opportunity assessment and associated risk mitigation and opportunity handling plans

   4) A SAR team assessment that the SRS, IRS, STP, SAD, and software operational concept document form a satisfactory basis for proceeding with software preliminary design

   5) A SAR team assessment that the software requirements and logical architecture baselines can be implemented within constraints of the budget, schedule and performance requirements

   6) Official, approved SAR minutes incorporating at a minimum the following items as part of the official record:

      i) All specifications, descriptions, plans, and analysis reports that were reviewed by the SAR team

      ii) A complete list of SAR attendees

      iii) Completed action item forms

      iv) Review results for each of the work product categories assessed during the technical review

      v) Configuration identification documentation that completely defines the software requirements and logical architecture baselines that are the object of the SAR work products

A.5.6 SAR exit criteria

The SAR shall be deemed completed only after the following events have been successfully completed:

a) All action items submitted during the technical review have been appropriately resolved.

b) All actions listed in the action items required for closure have been completed and approved by the required parties.

c) The content of corrective action plans, if any, for issues identified in the SAR is sufficiently complete and unambiguous to enable successful completion of all corrective actions.

d) Each of the technical review products listed in Table A.2 as tailored for the specific program meets all of its acceptability criteria, or has a corrective action plan documenting the corrective actions required to achieve acceptability.

e) The acquirer and supplier concur that the risk level is acceptable.

f) All baselines in the applicable configuration management (CM) system(s) are current and consistent with the audited SAR work products.

g) The SAR chair formally closes the review.

A.6 SAR detailed criteria

A.6.1 SAR review products acceptance criteria

Table A.2 lists the products that should be reviewed at the SAR, and the associated acceptability criteria that should form a sufficient basis by which to assess each product's content as acceptable to support a successful SAR. If a given program requires a SAR, specific products and associated acceptability criteria shall be deleted, modified, or additional items included, in accordance with the acquirer-supplier agreement, to support a given program.

Table A.2—SAR technical review products acceptability criteria

Product	SAR acceptability criteria
Software technical documentation	a) The higher-level requirements allocated to software are complete, consistent, and stable. b) Software requirements including interface requirements have been specified to the level of completeness called for in the SDP based on the selected software life-cycle model. c) Software requirements are correct, complete, consistent, feasible, verifiable, and unambiguously stated. d) Software requirements include necessary requirements derived from the system and software logical architecture, system operational concepts, trade studies, and design decisions. e) Each software requirement and interface requirement has one or more valid verification methods and levels specified, and those methods and levels are sufficient to fully verify the requirement. f) Software operational concepts include nominal and off-nominal scenarios, including workloads, for all defined system states and modes that are consistent with the system and software logical architectures. g) Software operational concepts include information exchange with external interfacing systems and are consistent with system operational concepts. h) The software logical architecture has been defined to the level of completeness called for in the SDP based on the selected software life-

Product	SAR acceptability criteria
Software technical documentation *(continued)*	cycle model and the architectural views, including physical, logical developmental process and behavioral are correct, complete, consistent, and unambiguous. i) The software logical architecture has been elaborated to the lower level of detail sufficient to address implementation of all allocated requirements. j) The software logical architecture satisfies the functional and performance requirements allocated to each state and mode. k) The software physical architecture adequately addresses the selected COTS software products, unmodified reuse software, modified reuse software, newly-developed software, programming language(s) to be used, and installation and configuration design decisions. l) The software physical architecture fully integrates any reusable software items (e.g., COTS, government off-the-shelf, modified and unmodified software, and NDI software), and will enable all software requirements to be met, including instances where reusable software elements are directed by the customer, which may result in a constraint to the architecture. m) The software logical architecture adequately addresses all external and internal interfaces including human-system interactions. n) The software logical and physical architectures adequately address all applicable standards. o) The software logical architecture adequately addresses end-to-end processing including mission timelines and all interoperability requirements. p) The software logical architecture adequately addresses the high-level design of all files, shared memory, operational database management, and storage and access methods. q) The software logical architecture adequately addresses operational database management and control. r) The software logical architecture adequately addresses supportability, including integrated hardware-software diagnostics, fault detection, isolation, localization, and repair. s) The software logical architecture adequately addresses R&M requirements allocated to the SWCIs. t) Computing resources have been selected and appropriately incorporated into the top-level system hardware and software physical architecture and will enable all software requirements to be met. u) Engineering analyses, models, and simulations adequately demonstrate that the software physical architecture addresses safety, cybersecurity, human systems integration, and use of reusable software. v) Engineering analyses, models, and simulations adequately demonstrate that the software physical architecture together with the selected computer resources will meet the key performance parameters (KPP). w) A preliminary performance analysis at the current point in the development life cycle indicates that the software physical architecture together with the selected computer resources has an acceptable likelihood of meeting the performance requirements with adequate margins. x) Engineering analyses and trade studies for human systems integration demonstrate the adequacy of the software physical architecture and computer resources that have been selected for the operators to perform their required roles within the required timelines. y) Software qualification test plans have been defined to the level of completeness called for in the SDP based on the selected software life-cycle model. z) Software qualification test plans are valid, complete, stable, and consistent with the software physical architecture and higher-level test plans.

Product	SAR acceptability criteria
Software technical documentation *(continued)*	aa) All software requirements are allocated to the tests described in the software qualification test plans where they will be verified. bb) The software master build plan is complete, feasible, executable, and consistent with the software requirements, software physical architecture, and software qualification test plans. cc) All traceability information is bi-directional, complete, and consistent with the software requirements, software logical architecture elements, and software qualification test plans, and the parent requirements.
Technical plans	a) The SDP is consistent with the integrated master plan (IMP), SEMP, and any other software-related engineering and management plans. b) The SDP fully describes the selected software development life-cycle model(s) that are feasible, appropriate for program scope and complexity, and used consistently by all team members. c) The SDP includes an integrated set of effective processes, methodologies, tools, and environments that cover all software team members, are suitable for the domain, and are appropriate for program scope and complexity. d) The SDP adequately addresses all applicable software standards, processes, procedures, and conventions. e) The SDP includes a sound software risk management plan that is integrated with the program risk management plan. f) Effective software risk handling plans (mitigation as well as other handling methods) are in place, and risk handling activities are being performed in accordance with the plans.
Program execution and process control	a) The supplier has demonstrated that the software processes, standards, procedures, and conventions are being followed as appropriate for the current point in the life cycle. b) An effective program-level risk management process including the software risk management process has been demonstrated to be effectively functioning. c) The program's defined software measures and metrics are sufficient for meeting the program's information needs, and are being collected, analyzed, reported, and used for management and technical decision-making as appropriate at the current point in the life cycle. d) Adequate corrective actions have been defined to address the underlying problems indicated by software metrics that are outside of documented thresholds. e) The existing and planned software engineering environment (SEE) is integrated with the system engineering environment across all software team members for the software under review. f) The SEE and the software test environment(s) have adequate capability and capacity to meet the software development and test requirements and schedules. g) Technical performance measures (TPM) are being collected, analyzed, reported, and used for managing the software-related KPPs and utilization of all critical computer resources. h) Adequate corrective actions have been defined to address the underlying problems indicated by software TPMs that are outside of documented thresholds. i) The supplier has demonstrated that corrective actions have been initiated, managed, and tracked to closure for metrics or TPMs that are outside of documented thresholds. j) The program's software problem and deficiency reporting system is operational and has demonstrated its effectiveness in implementing and verifying solutions to documented problems, prioritized by their severity.

Product	SAR acceptability criteria
Risk assessment	a) Technical risks are identified, and mitigation plans in place. b) Risk management process is in place demonstrated by documented execution results of existing mitigation plans associated with the purpose of the review.
Program cost and schedule estimates	a) Software cost models have been calibrated with realistic software cost drivers based on actual data, both for the current program and past history, and are used to update software cost and schedule estimates. b) Software size estimates are supportable based on history, are consistent with the software requirements and software physical architecture, and have sufficient margins to cover the estimated risk at the current point in the life cycle. c) Software schedule items have been integrated into the IMS with critical path dependencies identified. d) Software resources planned for sustainment are consistent with the progress and rate of development.

A.6.2 SAR preparation

Table A.3 lists the actions that should be considered during preparation for the SAR. The responsible people listed are those most often tasked with the listed preparation actions, but the acquirer and supplier may agree to assign the actions to different people or organizations depending on a given program's organizational structure. If a given program requires a SAR, the specific actions shall be deleted, modified, or additional items included, in accordance with the acquirer-supplier agreement. If a given program requires a SAR, responsibilities shall be assigned to people or organizations in accordance with the acquirer-supplier agreement.

Table A.3—SAR technical review preparation actions

Responsible person	SAR preparation actions
Program manager	a) Approve, fund, and staff the SAR as planned in the Systems Engineering Plan (SEP) developed by the systems engineer. b) Appoint a SAR chair no later than 45 days prior to the technical review, in coordination with the systems engineer and program lead software engineer. c) Coordinate a preliminary agenda between the program integrated product team (IPT) and other acquirer subject matter experts (SME), no later than 30 days prior to the SAR.
Systems engineer	a) Ensure adequate plans are in place to complete the technical activities to proceed from SAR to PDR. b) Ensure all of the technical review products whose acceptability criteria are defined in Table A.2 are completed for the SAR.
Program lead software engineer	a) Coordinate arrangements for SAR location and support. b) Coordinate requirements for the SAR chair with the systems engineer and program manager. c) Coordinate the preliminary SAR agenda with the systems engineer and program manager. d) Ensure the preparation of all presentation material is coordinated across IPTs.
SAR chair	a) Determine SAR team membership. b) Approve the final SAR agenda. c) Approve any final Clause A.6 SAR detailed criteria tailoring for the specific program. d) Identify any specific elements for in-depth technical review as required.

A.6.3 SAR conduct

Table A.4 lists the technical review elements and associated content details that should be considered for the conduct of the SAR. If a given program requires a SAR, specific elements and their content details shall be deleted, modified, or additional items included, in accordance with the acquirer-supplier agreement, to support a given program.

Table A.4—SAR conduct elements

SAR review element	Content details
Software technical documentation review	a) Requirements: 1) Parent requirements allocated to software 2) Software functional, performance, non-functional, and interface requirements 3) Software-related external and intersegment or element interface requirements. 4) Interface control documents with software-to-software and software-to-hardware interface requirements b) Software operational concepts c) Software architectural design: 1) Software architectural design description 2) Top-level computer system hardware-software architectural design description d) Engineering analyses: 1) Software engineering analyses, trade studies, modeling, and simulation results 2) Hardware-to-software engineering analyses, hardware vs. software trade studies, integrated hardware-software M&S results e) Integration and verification: 1) Software master build plan (allocation of requirements, functionality, and architectural components to builds) 2) Software qualification test planning f) Traceability: 1) Bi-directional traceability between higher-level requirements allocated to software (including interface requirements) and software requirements (including interface requirements) 2) Bi-directional traceability between software requirements (including software interface requirements) and software logical architecture elements 3) Bi-directional traceability between software requirements (including software interface requirements) and software qualification tests 4) Bi-directional traceability among requirements and verification methods and verification integration levels 5) Bi-directional traceability among builds and software requirements, software logical architecture elements, and software qualification tests

SAR review element	Content details
Technical plans review	a) SDP structure, content, consistency with chosen software development life-cycle model b) Other software-related program plans (e.g., system engineering management plan, risk management plan, integrated master plan, CM plan, quality assurance plan) c) System-level (hardware and software) specialty engineering plans (e.g., reliability, safety, supportability, security, cybersecurity, human systems integration) d) Plans and status of the software engineering environment (SEE) e) Plans and status of the software test environments, including test beds, test facilities, hardware, software, simulators, and other testing tools
Program execution and process control review	a) Program change control system and CCB organization and operation b) Software problem and deficiency report status c) Software measures, metrics definitions, and status reports d) Software TPM reports
Risk assessment review	c) Risk identification and mitigation, including consideration of the following: 1. Software requirements, size, and complexity 2. Selection and use of reusable software 3. Population, update, control, and validation of databases 4. System and software architectural choices 5. Computing resources, including margins 6. Computer hardware and software technology 7. Software schedules 8. Software development, integration, and verification processes and tools
Program cost and schedule review	a) Software size, effort, cost, and staffing estimates b) Software schedules c) Higher-level schedule(s) including the IMS d) Initial and final developer report, and data dictionary for applicable builds e) Updates to the life-cycle cost (LCC) and cost as an independent variable (CAIV) studies presented at the system requirements review (SRR) in support of each software logical architecture. For example: 1) LCC and CAIV modeling and analyses as they are applied and correlated with cost models depicting projected program development, operational and sustainment costs completed, as well as projected cost impacts to other external systems 2) LCC and CAIV methodologies addressed by trade studies

A.6.4 SAR closure

Table A.5 lists the actions that should be considered for SAR closure. The responsible people listed are those most often tasked with the listed closure actions. The acquirer and supplier may agree to assign the program manager and systems engineer actions to different people or organizations depending on a given program's organizational structure. If a given program requires a SAR, the specific actions shall be deleted, modified, or additional items included, in accordance with the acquirer-supplier agreement. If a given program requires a SAR, responsibilities shall be assigned to people or organizations in accordance with the acquirer-supplier agreement.

Table A.5—SAR closure actions

Responsible person	SAR closure actions
Program manager	a) If funding profiles are insufficient to support development, notify user/sponsor of funding shortfall and request funding profile adjustments. b) Support development of the SAR summary report.
Systems engineer	a) Organize and supervise the responses to all action items generated during SAR. b) Support development of the SAR summary report.
Program lead software engineer	a) Organize and supervise the detailed documentation of all action items assigned during the SAR. b) Support development of the SAR technical review summary report.
SAR chair	a) Ensure preparation of the SAR summary report with the support of the program manager, systems engineer, and program lead software engineer. b) Sign off final approval of all action items. c) Prepare the formal SAR completion letter.
Recorder	a) Collate all action items for submission to the SAR chair. b) Prepare the SAR summary report and SAR minutes for signature and distribution by the SAR chair. c) Prepare the SAR closure letter for signature by the SAR chair.

A.7 SAR application guidance

The following is a set of observed good practices for consideration:

a) The SAR may be conducted in place of the software specification review (SSR) if the software logical architecture is sufficiently mature, and it is consistent with the selected software life cycle as defined in the SDP.

b) The scope of the SAR should be tailored to be consistent with the technical scope of the overall program.

c) The request for the SAR chair should occur at least 60 days prior to conduct of the technical review.

d) The positioning of the SAR in the software development life cycle is dependent upon the life-cycle model in use for the software under review:

   1) In the waterfall life-cycle model, the software is developed in a "once through" fashion where the sequence of software requirements definition, software architectural and detailed design, software implementation and software testing occurs only once. For the waterfall life-cycle

model the SAR is conducted at the completion of the software architectural (high-level) design.

2) In an incremental life-cycle model, the software requirements are *defined* first. Then, the software is developed in a series of builds where each build adds to the previous build and enhances its capabilities. In the incremental life-cycle model, each build consists of a once-through sequence of software requirements *assessment*, software architectural and detailed design, software implementation and software testing. For the incremental life-cycle model, the SAR is conducted iteratively at the conclusion of the software logical and physical architecture design for each incremental build. The full set of software requirements for the software under review is reviewed in the build-1 SAR, while the software logical and physical architectures and other information are reviewed in an incremental fashion as each build proceeds.

3) An evolutionary life-cycle model is similar to the incremental life-cycle model, with the exception that, in the evolutionary life-cycle model, each of the builds is based upon the requirements allocated to the software build from the parent specification. In the evolutionary life-cycle model, each build consists of a once-through sequence of software requirements *definition*, software architectural and detailed design, software implementation, and software testing.

d) The software life-cycle model(s) in use, the positioning of the SAR(s) within the life-cycle model(s), and the relationship of the positioning of the SAR(s) with respect to the other major reviews defined in this standard as tailored for the specific program, should be defined in the SDP.

e) In order to help ensure a comprehensive and balanced assessment of all SAR work products, SAR participants from both the acquirer and supplier should include the following, as applicable:

1) Program management
2) Configuration management
3) Systems engineering
4) Software engineering
5) System safety
6) Logistics
7) Test and evaluation
8) All certification authorities
9) System users
10) Cost estimating team
11) Legal counsel, if required
12) Contracting officers
13) Recorder or secretary

NOTE—These roles do not dictate that a single individual is provided for each role. A single individual may perform more than one of these roles within the team. Depending on the complexity of the system, more than one individual may also be assigned to a specific role.