A DoD-Oriented Introduction to the NDIA's System Assurance Guidebook

Paul Popick
The Aerospace Corporation

Dr. Terence E. Devine
The MITRE Corporation

Rama Moorthy
Hatha Systems

Despite significant strides toward building secure and trustworthy systems, there is ample evidence that adversaries retain their ability to compromise systems. "Engineering for System Assurance" [1] *(the Guidebook)—a publication from the National Defense Industrial Association (NDIA)—provides process and technology guidance to increase the level of systems assurance (SA). One section has guidance particularly useful to the DoD and their contractors. This article provides an introduction to key SA activities with an emphasis on the selected reviews within the DoD Life-Cycle Management Framework.*

For decades, industry and defense organizations have tried to build affordable, secure, and trustworthy systems. Despite significant strides toward this goal, there is ample evidence showing that adversaries retain their ability to compromise systems. Consequently, there is growing awareness that systems must be designed, built, and operated with the expectation that system elements will have both known and unknown vulnerabilities.

SA is defined as:

> The justified confidence that the system functions as intended and is free of exploitable vulnerabilities, either intentionally or unintentionally designed or inserted as part of the system at any time during the life cycle. [1]

This ideal of no exploitable vulnerabilities is usually unachievable in practice, so programs must perform risk management to reduce (to acceptable levels) the probability and impact of vulnerabilities.

This confidence is achieved by SA activities, which include a planned, systematic set of multi-disciplinary activities to achieve the acceptable measures of SA and manage the risk of exploitable vulnerabilities. The *assurance case* is the enabling mechanism showing that the system will meet its prioritized requirements and that it will work as intended in the operational environment, minimizing the risk of exploitation through weaknesses and vulnerabilities.

The Guidebook is intended primarily to aid program managers and systems engineers seeking guidance on how to incorporate assurance measures into their system life cycles. Assurance for security must be integrated into the systems engineering activities to be cost-effective, timely, and consistent. The activities for developing and maintaining the assurance case enable rational decision-making so that only the actions necessary to provide adequate justification (arguments and evidence) are performed. The Guidebook is a synthesis of knowledge gained from existing practices, recommendations, policies, and mandates. SA activities are executed throughout the system life cycle. It is organized based on the ISO/IEC's "Systems and Software Engineering – System Life Cycle Processes" [2]; while there are other life-cycle frameworks, this standard combines a suitably encompassing nature while also providing sufficient specifics to drive SA.

> *"... programs must perform risk management to reduce (to acceptable levels) the probability and impact of vulnerabilities."*

This Guidebook also provides an assurance guidance section for use by the DoD and their contractors and subcontractors. Future editions of this Guidebook may add additional domain-specific assurance guidance.

This article provides an overview of the assurance case section (Section 2.2) and then describes key SA activities completed for selected reviews within the DoD Integrated Defense Acquisition, Technology, and Logistics Life-Cycle Management Framework (referred to in this article simply as the DoD Management Framework) from Section 4 of the Guidebook.

Assurance Case[1]

The purpose of an assurance case is to provide a convincing justification to stakeholders that critical SA requirements are met in the system's expected environment(s). Any assurance claims about the system need to be incorporated into the system requirements.

An assurance case is the set of claims of critical SA properties, arguments that justify the claims (including assumptions and context), and evidence supporting the arguments. The development of the assurance case results in SA requirements that are then flowed to the system architecture and the product baseline. The assurance case can be considered an extension or adaptation of the safety case, which has been used for safety-critical systems. Thus, the concept is not entirely new.

The assurance case need not be a separate document; it may be distributed among or embedded in existing documents. Even if there is an assurance case document, it would typically contain many references to other documents. Regardless of how the assurance case is documented, there must be a way to identify all of the assurance claims, and from those claims trace through to their supporting arguments, and from those arguments to the supporting evidence. For example, an organization might maintain a list of system requirements, tagging specific ones as assurance claims with hyperlinks to arguments that justify why the system will meet the claim. As a minimum:

1. Assurance case claims, arguments, and evidence must be relevant for the system and its operating environment(s).
2. Claims are justified by their arguments.
3. Arguments are supported by their evidence.
4. The assurance case must be developed iteratively, be sustainable, and be maintained throughout the system life cycle

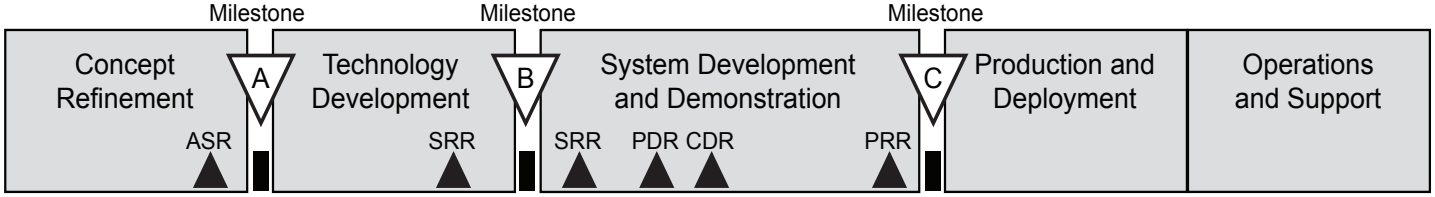

Figure 1: *Life-Cycle Phases with Reviews (from the 2003 Version of DoD Instruction 5000.2)*

as a living document.

5. The assurance case must be delivered as part of the system, to be maintained during system sustainment.

The Guidebook makes no attempt to specify a format for an assurance case, only providing guidance as to what information should be included. The current revision of ISO/IEC 15026 [3] specifies a standard for assurance cases.

The assurance case is generated by the systems engineering technical activities applied to the assurance requirements, and provides evidence of the growing technical maturity of the integration of the SA requirements for use within event-driven technical management. Sections 3 (general) and 4 (DoD-specific) of the Guidebook relate the assurance case to the life-cycle processes.

Figure 2: *ASR Excerpt*

To successfully complete the ASR review, ensure the following SA items were satisfactorily completed:

- System threats and SA claims were considered as part of the analysis of alternatives and full system life-cycle costs were used in the analysis. Sometimes the seemingly cheapest alternative has higher system life-cycle costs due to assurance complications.
- A preliminary identification of critical technologies with a description of how to assure these technologies. This list will eventually feed the Critical Program Information (CPI) developed at the start of the System Development and Demonstration phase. As an aid to this identification, review the Military Critical Technologies List (http://www.dtic.mil/mctl/).
- The Initial Capabilities Document (ICD) and Preliminary System Specification includes:
  - The requirement for the development of an assurance case with high-level claims for each system element determined to be critical.
  - The sustaining mission operational requirements constrain the top-level SA claims to counter identified threats to the mission. They should broadly identify an approach for developing the system assurance case.
  - A critical elements list.
  - The Support and Maintenance Concepts and Technologies with a description of how assurance will be maintained.

Figure 3: *SRR Excerpt*

To successfully complete the SRR, ensure that the following SA items were satisfactorily completed:

- ICD and Preliminary System Performance Specification:
  - Establishes the requirement for the development of an assurance case, including high-level claims for a system determined to be critical.
  - The operational requirements necessary to sustain the mission include the top-level SA claims that address identified threats to the mission that are the foundation for the assurance case. Critical elements are identified.
  - The Support and Maintenance Concepts and Technologies documented with a description of how assurance will be maintained.
  - Requirements with SA implication have been tagged for SA traceability and verification.
- Identification of all critical elements to be protected, and what aspects of them are to be protected (e.g., confidentiality, integrity, availability, authentication, accountability [including non-repudiation], and auditability). For example, ensure that an adversary cannot gain control over a weapon system.
  - Initial identification of potential CPI, and a preliminary approach to the protection of that CPI is part of the system requirements.
  - Identification of all relevant SA threats and their potential impact on critical system assets.

The DoD Management Framework

Section 4 is for use by the DoD and DoD contractors and subcontractors. It is organized according to phases of the DoD Management Framework discussed in DoD Directive 5000.1 [4], DoD Instruction 5000.2 [5][2], and the Defense Acquisition Guidebook (DAG) system life cycle [6].

Section 4's goal is to enable the DoD to acquire or produce assured systems, including both weapons systems and IT systems. This section discusses the topics that should be addressed during the phases of the DoD Management Framework. As discussed in [5] and [6], the life-cycle phases include:

- Concept Refinement.
- Technology Development.
- System Development and Demonstration.
- Production and Deployment.
- Operations and Support.

Section 4.3 is subdivided into DoD review milestones and milestone decision points in this framework. For each review (including the System Engineering Technical Reviews), the DAG description is quoted, followed by a list of the most important SA items to complete prior to that milestone. The non-SA activities normally associated with the review are not specified, but should be considered as the context in which the SA activities are performed. For each review, a specific cross-reference to the corresponding general technical instruction is also provided. Where there are assurance activities that cannot be associated with specific reviews, additional subsections are added to contain them.

Figure 1 depicts the DoD life-cycle phases. The reviews that are discussed in this article are shown in their phases to provide a visualization of when the review occurs. Note that the reviews shown are a subset of life-cycle reviews.

Section 4 focuses on the DoD and makes the assumption that the audience includes system integrators providing systems (both IT and warfighting) to the

DoD environment.

The following subsections provide a description of selected key activities for a subset of the reviews and events.

Alternative Systems Review (ASR)

The ASR occurs during the Concept Refinement Phase prior to Milestone A. Figure 2 shows a partial excerpt of the list of the SA activities to be completed prior to the ASR.

At this point in the life cycle, a key SA activity for each alternative is shown in the first bullet of Figure 2. For each alternative, a threat analysis and the assurance case claims (to counter the identified threats) need to be developed. This analysis needs to be factored into the alternative cost estimates, the selection of a preferred system concept, and the technology development strategy. SA is made more complex when it is not considered in the early stages when alternative concepts are being evaluated.

Systems Requirements Review (SRR)

The SRR can occur at the end of the Technology Development Phase prior to Milestone B, at the start of the System Development and Demonstration Phase, or both. Figure 3 shows a small excerpt of the list of the SA activities to be completed prior to the SRR.

Development of the assurance case claims assists with the development of the system requirements for assurance. Using an assurance case that systematically examines the claims necessary to counter the threats will produce a set of derived security requirements that can be added to the system requirements. Having a robust set of security requirements by SRR allows the security to be built into the system rather than tacked on during the testing or production phase.

Preliminary Design Review (PDR)

The PDR occurs during the System Development and Demonstration Phase, after Milestone B[3]. Figure 4 shows an excerpt from the list of the SA activities to be completed prior to the PDR.

The first bullet of Figure 4 lists the identification of critical components and examination of those components for weaknesses and vulnerabilities. Critical components may be managed through techniques such as graceful degradation, isolation, modularity, diversity, single-point-of-failure reduction/multipathing, and the use of interchange standards to reduce the number, size, and impact of critical elements. Many of these approaches become cost-prohibitive if the weakness and vulnerability analysis is delayed until late-stage testing and production. These activities tie into IA through its control for vulnerability management, known as VIVM-1.

To successfully complete the PDR, ensure the following SA items were satisfactorily completed:

- Use the architecture and preliminary design information (as available) to identify critical components. Identify weaknesses and their associated potential vulnerabilities. Note that a weak architecture can result in a systemic weakness, which can in turn lead to many vulnerabilities. Thus, a rigorous review of the architecture may be required prior to release to design phase. Refine and document a baseline of attack scenarios of the identified threats and assets (see Information Assurance [IA] control: VIVM-1).
- Development of specific instances of SA scenarios, at least for the critical SA requirements, to verify that the system will counter the attack.
- Architecture and preliminary system design includes IA accreditation requirements in its relationship to all hosting enclaves and impact analysis is completed for the architecture. Develop a list of all hosting enclaves as a baseline for tracking purposes as the system moves into the detailed design phase (see IA controls: DCII-1, DCID-1).
  - Ensure that the architecture and preliminary system design of mobile code usage is evaluated for acceptable risk, avoiding high risk as defined by DoD requirements (see IA control: DCMC-1 [Mobile Code]).
  - Exclude binary or machine executable public domain software products with no warranty and no source code (see IA control: DCPD-1).

Figure 4: *PDR Excerpt*

Critical Design Review (CDR)

The CDR occurs during the System Development and Demonstration Phase. Figure 5 shows an excerpt from the list of the SA activities to be completed prior to the CDR.

The first bullet in Figure 5 indicates that prior to the CDR the system requirements, the functional baseline, and the allocated baseline need to be updated to incorporate the claims, arguments, scenarios, and any design changes as a result of the assurance case analysis. The fourth main bullet indicates the need to define and select assurance-specific static analysis and criteria for examination during peer reviews (performed during implementation). These are key activities needed to

Figure 5: *CDR Excerpt*

To successfully complete the CDR review, ensure the following SA items were satisfactorily completed:

- Update the system requirements, the functional baselines, and the allocated baseline to incorporate the claims, arguments, and scenarios.
- Capture assurance designs in the associated configuration item build-to documentation as part of the system's Product Baseline. The system's configuration item verification planning should be updated and included in the Product Baseline.
- Update the SA case based on the design, new weaknesses, vulnerabilities identified, and the preceding analysis.
  - For each SA claim, define the detailed argument(s) to be used to justify the claim, identify the expected evidence (type and expected measure) that will support the argument, and how that evidence will be acquired (including what verification data must be created to acquire that evidence).
  - After CDR, the assurance case's claims and argument structure are baselined, some evidence is already available, and the methods for acquiring the remaining evidence have been defined.
- Define and select assurance-specific static analysis and assurance-specific criteria to be examined during peer reviews, to be performed during implementation.
  - Plan for training for assurance-unique static analysis tools and peer reviews.
  - Ensure that another party (such as a peer) will independently perform static analysis and test, and that the element being reviewed will be the element that will be delivered. This counteracts the risk of a developer intentionally subverting analysis and test, as well as aiding against unintentional errors.

To successfully complete the PRR review, ensure the following SA items were satisfactorily completed:

- Incorporation of the results of system test for weaknesses and their associated vulnerabilities into the assurance case. Verification that the system weaknesses and vulnerabilities have been baselined and appropriately documented (see IA control: VIVM-1).
- Verification that the developed system uses comprehensive test procedures to test any and all patches and upgrades required throughout its life cycle (see IA control: DCCT-1).
- Incorporation of the results of any testing, using industry tools and test cases, for any binary or machine-executable public domain software products with no warranty and no source code being used in the system (see IA control: DCPD-1).
- Evaluation of the relevant test results to obtain the evidence required to build the assurance case from the following list of defensive functions of a system as well as assurance mechanisms that address security, partitioning, access, and traceability mechanisms:
  - Evaluation of the protection mechanisms with each external interface and associated security requirements using test results as well as other evidence (see IA control: DCFA-1).
  - Evaluation of the adequacy of security best practices of such functions as identification/authentication (individual and group) using DoD PKI, logon including single sign-on, PKE, key management, smart card, and biometrics (see IA controls: as per DoDI 8500.2, DCBP-1, IATS-2, IAAC-1, IAGA-1, IAIA-2, IAKM-3, ECLO-2, ECPA-1).
  - Reverification that user interface services are logically or physically separated from data storage and management services. This is particularly important in high assurance systems (see IA control: DCPA-1).

Figure 6: *PRR Excerpt*

ensure the software being developed. Also at this point, the design must meet IA accreditation requirements, including:

- Developing a list of hosting enclaves.
- Evaluating mobile code usage.
- Excluding binary and machine-executable public domain software products with no warranty and no source code.

Production Readiness Review (PRR)

The PRR examines whether the system is ready for production. From an SA standpoint, the test results are examined to ensure that all of the system requirements have been met. This entails looking at test results that substantiate the claims in the assurance case. Figure 6 shows an excerpt from the list of the SA activities to be completed prior to the PRR.

The first bullet discusses the need to incorporate test results for weaknesses into the assurance case as evidence. The weaknesses and vulnerabilities need to be baselined and documented appropriately in accordance with the applicable IA control. The extent to which SA work was performed during the early stages is apparent during the verification of the test results tied to each of the requirements; early emphasis greatly simplifies the verification.

Conclusion

The Guidebook is intended to help alleviate problems by increasing awareness of SA issues, encouraging these issues to be addressed early in the development life

About the Authors

Paul Popick is the associate director of software systems engineering with the Aerospace Corporation. He has more than 35 years of experience in project management and systems engineering management. Popick is a principal author of the NDIA's "Engineering for System Assurance" Guidebook. Prior to joining the Aerospace Corporation, he was director of delivery excellence for a unit of IBM Global Services.

Aerospace Corporation
15049 Conference Center DR
MS CH3-320
Chantilly, VA 20151
Phone: (571) 307-3701
Fax: (571) 307-1833
E-mail: Paul.R.Popick@aero.org

Terence E. Devine, Ph.D., is a software engineer with MITRE. He has more than 35 years of experience in software design and development. Devine is a principal author of the NDIA's "Engineering for System Assurance" Guidebook. He has a doctorate in computer science from UCLA.

The MITRE Corporation
260 Industrial Way West
Eatontown, NJ 07724
Phone: (732) 578-6339
Fax: (732) 578-6014
E-mail: tdevine@mitre.org

Rama Moorthy, CEO of Hatha Systems, has more than 20 years of experience in the high-tech industry, building and delivering products, services, and strategies—both in the commercial and government sectors. In addition to her role as CEO, she supports the DoD's Globalization Task Force on software assurance and supply chain risk management. Moorthy is a principal author of the NDIA's "Engineering for System Assurance" Guidebook. She has a bachelor's degree in electrical engineering and an MBA (marketing and finance).

Hatha Systems
1101 Pennsylvania AVE, NW
STE 600
Washington, D.C. 20004
Phone: (202) 756-2974
E-mail: Rama.Moorthy@hathasystems.com

cycle, and providing pointers to available resources. The Guidebook shows how SA can be implemented in the existing environment and life cycle. It does not represent original research, but rather a survey of existing work.

The plan is to update the Guidebook to reflect and incorporate feedback received from the programs that use it, December 2008 changes to [5], and new techniques as they emerge. The expectation is that this article will encourage programs to use the Guidebook and to address SA issues early in the life cycle.◆

Acknowledgements

The authors would like to thank the following:

- The NDIA SA Committee Chairs: Kristen Baldwin, Office of the Under Secretary of Defense (Acquisition, Technology, and Logistics); Mitch Komaroff, Office of the Assistant Secretary of Defense (Network and Information Integration); Paul Croll, Computer Sciences Corporation.
- The other principal authors of the Guidebook: David Wheeler, Marie Stanley Collins, Murray Donaldson, and John Miller.
- Arch McKinlay, who wrote the original draft Guidebook's Assurance Case section, which is quoted extensively in this article.

References

1. NDIA – Systems Assurance Committee. *Engineering for System Assurance*. Oct. 2008 <www.acq.osd.mil/sse/docs/SA-Guidebook-v1-Oct2008.pdf>.
2. ISO/IEC. *Systems and Software Engineering – System Life Cycle Processes*. International Standard 15288-2008. 1 Feb. 2008.
3. ISO/IEC. *Systems and Software Engineering – Systems and Software Assurance*. International Standard 15026. Draft.
4. DoD. *The Defense Acquisition System*. Directive 5000.1. 2003.
5. DoD. *Operation of the Defense Acquisition System*. Directive 5000.2. 12 May 2003.
6. Defense Acquisition University. *Defense Acquisition Guidebook*. 12 Dec. 2004.

Notes

1. The Introduction and Assurance Case sections of this article are excerpted from [1].
2. The Guidebook is written the 2003 release of 5000.2 [5]; although the current release (2008) has changed the phases, the Guidebook 5A instructions for the systems engineering technical review criteria remain applicable.
3. The 2008 release of DoD Instruction 5000.2 now has the PDR prior to Milestone B.

Additional Resources

1. DoD. *Information Assurance*. Directive 8500.01E. 23 Apr. 2007 <www.dtic.mil/whs/directives/corres/pdf/850001p.pdf>.
2. DoD. *Acquisition Systems Protection Program*. Directive 5200.1-M. Mar. 1994 <www.dtic.mil/whs/directives/corres/pdf/520001m.pdf>.
3. DoD. *National Industrial Security Program*. Directive 5220.22. 1 Dec. 2006 <www.dtic.mil/whs/directives/corres/pdf/522022p.pdf>.
4. DoD. *Defense Intelligence Agency*. Directive 5105.21. 18 Mar. 2008 <www.dtic.mil/whs/directives/corres/pdf/510521p.pdf>.
5. DoD. *Security, Intelligence, and Counterintelligence Support to Acquisition Program Protection*. Directive 5200.39. 10 Sept. 1997 <http://fas.org/irp/doddir/dod/d5200_39.pdf>.
6. DoD. *Operation of the Defense Acquisition System*. Instruction 5000.2. 8 Dec. 2008 <www.dtic.mil/whs/directives/corres/pdf/500002p.pdf>.
7. DoD. *Information Assurance Implementation*. Instruction 8500.2 <www.dtic.mil/whs/directives/corres/pdf/850002p.pdf>.
8. DoD. *DoD Information Assurance Certification and Accreditation Process*. Instruction 8510.01. 28 Nov. 2007 <www.dtic.mil/whs/directives/corres/pdf/851001p.pdf>.
9. Information Assurance Technology Analysis Center. *DoD Information Assurance and Computer Network Defense Strategies: A Comprehensive Review of Common Needs and Capability Gaps – State-of-the-Art-Report*. 21 July 2005 <http://iac.dtic.mil/iatac/pdf/gap.pdf>.
10. DHS. *Security in the Software Lifecycle: Making Software Development Processes—and the Software Produced by Them—More Secure*. Draft 1.2. Aug. 2006 <www.sis.uncc.edu/~seoklee/teaching/Papers/SwA%20Security%20in%20the%20Software%20Lifcycle%20v1.2%20-%20091306.pdf>.
11. Redwine, Samuel T., Jr., Ed. *Software Assurance: A Curriculum Guide to the Common Body of Knowledge to Produce, Acquire, and Sustain Secure Software*. Version 1.2. U.S. Department of Homeland Security. Oct. 2007 <https://buildsecurityin.us-cert.gov/daisy/bsi/resources/dhs/927-BSI.html>.