Meeting the Challenge of Assuring Resiliency Under Stress

Don O'Neill
Independent Consultant

An emerging issue, especially critical to the DoD and DHS, is that of managing network security, assuring the continuity of operations for critical defense missions and the resiliency of the private sector's critical infrastructure. Making systems of systems resilient requires accountability and transparency. This article provides a framework for assuring resiliency under stress expressed in terms of the management, process, and engineering indictors useful in asserting resiliency assurance claims, validating assurance arguments, and verifying assurance evidence.

Next generation software engineering faces many challenges [1], and the impacts of these challenges are being encountered every day by acquisition agents, software developers, and operating commands alike:

1. Acquisition agents need to deliver more with less ... fast.
2. Software developers need to shorten software development life cycles in producing trustworthy software systems composed of existing components.
3. Both acquisition agents and software developers need to exhibit better user domain awareness.
4. Operating commands need to field and sustain resilient systems of systems composed of legacy systems.

The industry has been grappling with many of these issues for years [2, 3]. Persistent acquisition challenges and chronic software development cost and schedule overruns frequently obscure the needs of the user. Despite this past neglect and unfinished business, the challenge of assuring resiliency under stress in systems of systems has emerged as an imperative that needs attention now.

In managing the investment needed to meet these objectives, capability portfolio investments are organized by management, process, and engineering. To receive results, utilize the objective (shown in Table 1) from top to bottom. In this way, user domain awareness, shortened life cycles, systems from parts, and systems of systems from systems provide a natural spiral of incremental activities where current work in progress builds on preceding work accomplished.

Resiliency Defined

The attribute of resiliency is an emerging property of large complex software-intensive systems. Accordingly, the base definition of resiliency is:

> ... the ability to anticipate, avoid, withstand, minimize, and recover from the effects of adversity, whether natural or manmade, under all circumstances of use. [4]

The base definition of resiliency is not limited as to scale, does not preclude the possibility for avoiding the condition or situation that brings impact or shock, does not limit the focus to a means like risk management, and does not limit the focus to enumerated outcomes like cost effective or timely restoration. However, in applying the base definition to a particular situation, it is permissible and even required to constructively instantiate it for targeted scale, impact expected, means employed, and outcome anticipated [5, 6].

Claiming Resiliency Assurance

The purpose of assurance assertion management is to reason about the emergent properties of large complex software-intensive systems in order to steer acquisi-

Table 1: *Practical Next-Generation Software Engineering (NGSE)*

Objective	Management Action	Process	NGSE Technology
Objective 1: *Drive user domain awareness towards more harmonious cooperation among people and machines.* **Strategic Measures:** 1. User satisfaction. 2. Trustworthiness.	**Integrate needs of systems, software, and user:** • Synthesize mission needs in terms of systems, software, and user. • Apply team innovation management.	**User domain awareness maturity:** • Assessment of user domain awareness.	• Simulation. • Virtual user experience.
Objective 2: *Simplify and produce systems and software using a shortened development life cycle.* **Strategic Measures:** 1. Speed. 2. Trustworthiness.	**Eliminate bottlenecks:** • Automation of labor-intensive activities.	**Accelerate delivery:** • Wiki-based requirements. • Incremental development. • Agile approaches.	• Formality in requirements expression. • Smart compilers. • Correctness by construction.
Objective 3: *Compose and field trustworthy applications and systems from parts.* **Strategic Measures:** 1. Frequency of release. 2. Trustworthiness.	**Rapid Release:** • Aspect-based commitment management. • Fact-based aspect and attribute assurance. • Real-time risk management.	**Supplier Assurance:** • Process maturity. • Global supply chain management. • Configuration management.	• Attribute-based architecture. • Smart middleware. • Interoperability. • Intrusion detection, protection, and tolerance.
Objective 4: *Compose and operate resilient systems of systems from systems.* **Strategic Measures:** 1. Control. 2. Resilience.	**Control:** • Exercise control.	**Awareness:** • Intelligent middlemen. • Information sharing. • Situation awareness.	• Coordinated recovery time objectives. • Distributed supervisory control. • Operation sensing and monitoring.

Overall Goal: *Drive systems and software engineering to do more with less ... fast.*

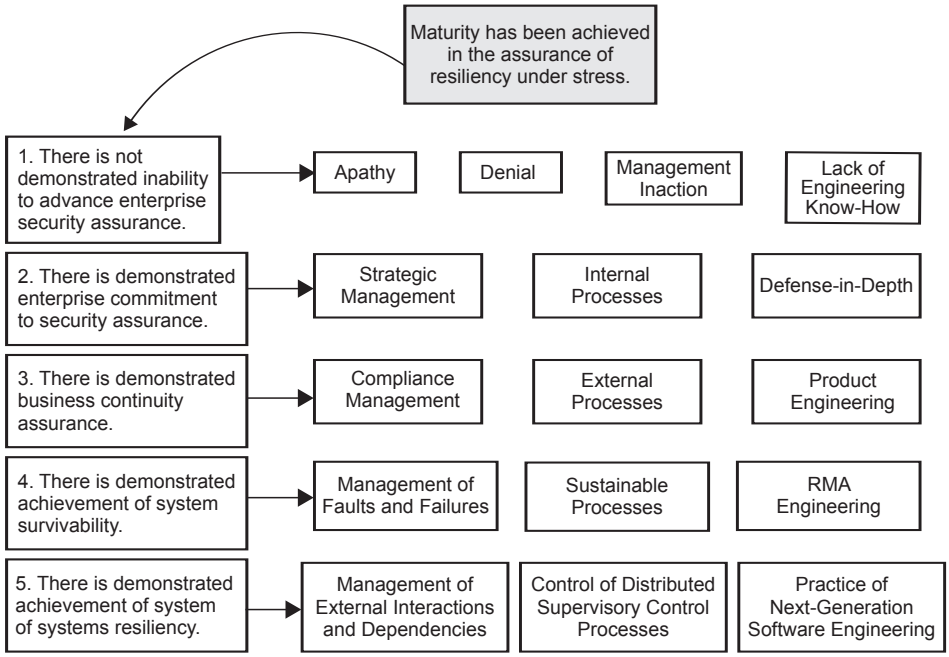


Figure 1: *Claim-Argument-Evidence Chain for Assessing Resiliency Assurance*

tion, development, and operational commitment towards their assurance and to guide users in setting the appropriate level of confidence in these systems and systems of systems [7].

An assurance assertion is a statement designed to inspire confidence. These emergent product properties transcend the rigorous and precise methods of assessing essential compliance beyond those used in process conformance [8, 9] and product testing. Some attribute and aspect examples of emergent properties associated with software products, systems, and system of systems include safety, security, resiliency, privacy, and trustworthiness [10].

The assurance claim for assuring resilience under stress in an enterprise is organized around five arguments expressed as questions:

1. Is there no demonstrated inability to advance enterprise security assurance?
2. Is there demonstrated enterprise commitment to security assurance through strategic management, internal processes, and defense-in-depth?
3. Is there demonstrated business continuity assurance through compliance management, external processes, and product engineering?
4. Is there demonstrated achievement of system survivability through the management of faults and failures, sustainability processes, and Reliability, Maintainability, and Availability (RMA) engineering?
5. Is there demonstrated achievement of system of systems resiliency through the management of external interactions and dependencies, the control of distributed supervisory control processes, and the practice of next generation software engineering?

The assurance claim for resilience assurance, the five arguments demonstrating resiliency assurance, and the types of evidence expected for each argument are shown in the Claim-Argument-Evidence Chain (Figure 1).

Assurance assertions themselves are subject to validation and verification, and it is here that managing the risk associated with assuring resiliency is focused. The claim-argument segment of the assurance assertion chain is validated when the correspondence between a claim and its arguments is shown to be clear and convincing with respect to completeness and correctness.

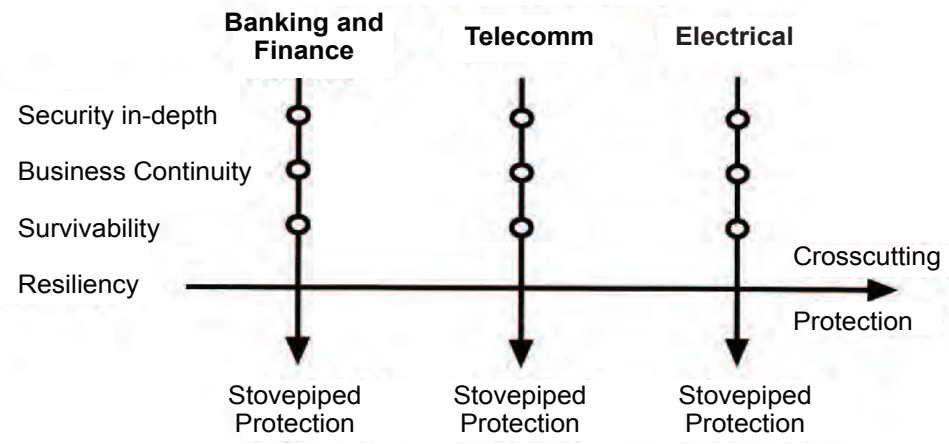
The argument-evidence segment of the assurance assertion chain is verified according to the degree of correspondence between the evidence and the argument. Four levels of confidence for appraising evidence are identified as follows:

1. The evidence in support of the argument is insufficient.
2. The preponderance of the evidence supports the argument (e.g., through assessment, interview, testimony, and inspection).
3. The evidence in support of the argument is clear and convincing (e.g., measurement and static analysis).
4. The evidence in support of the argument is beyond a shadow of a doubt (e.g., demonstration and dynamic analysis).

Achieving Resiliency

Assuring resiliency under stress is achieved through a framework of management, process, and engineering capabilities and indicators organized around managed review, defined process capability, and a designed engineering solution. Achieving system of systems resiliency brings with it an architectural challenge associated with the need to counter the effects of crosscutting and cascading triggers. Borrowing an example from the critical infrastructure and a dependency from the industrial base, stovepiped vertical protection, and crosscutting horizontal protection through resiliency are illustrated in Figure 2.

Figure 2: *Vertical Protection and Horizontal Resilience*



Crosscutting effects stem from dependent relationships. Some dependent relationships are planned and intended interactions between industry sectors—such as financial transactions embedded in telecommunications, electrical, transportation, and medical operations—where cross sector impacts are surprisingly pervasive [11]. Other dependent relationships are indirect and stem from outsourced commoditized services that bring with them opportunities for common single-point failures among industry sectors—

such as the Internet and global positioning systems [5].

Building on security in depth [12, 13, 14], business continuity [15], and system survivability [16], a defined engineering challenge of adopting system of systems resilience must be addressed [17]. The recovery time objectives among systems must be coordinated, interoperability of information sharing and platform operations must be assured, distributed supervisory control protocols must be in place, operation sensing and monitoring must be embedded, and digital situation awareness must be achieved. These capabilities are designed to counter crosscutting effects and cannot be expected to evolve in a loosely coupled environment. They must be holistically specified, architected, designed, implemented, and tested if they are to operate with resilience under stress [18]. A management, process, and engineering framework is necessary to advance the assurance of software security, business continuity, system survivability, and system of system resiliency capabilities (see Table 2).

Conclusion

This article has sought to point the way towards accountability and transparency in assuring the resiliency of systems of systems. Each operating command and critical infrastructure sector must insist on accountability from each system manager for its security in-depth, business continuity, and survivability. In addition, system managers must adopt transparency to the resiliency assurance claims, arguments, and evidence as the preferred means to achieve and demonstrate coordinated recovery time objectives, interoperability, operation sensing and monitoring, digital situation awareness, and distributed supervisory control.◆

Software Defense Application

The critical infrastructure is the industrial base on which the competitiveness and security of the nation are dependent. The defense industrial base finds itself in the mesh of the critical infrastructure. Diverse cybersecurity threats to the defense industrial base are posed by various factors parsed into type of risk, actor, attack, target, and countermeasure. For example, the type of actor includes disgruntled employee, hacker, criminal, terrorist, organized crime, and nation state. Faced with a complex array of threats, the critical infrastructure protection (CIP) model is insufficient to ensure the continuity of operations for critical missions. In addition to CIP, a critical infrastructure resiliency model is needed to anticipate, avoid, and mitigate cascading and propagating effects within systems of systems. "Meeting the Challenge of Assuring Resiliency Under Stress" provides a definition and framework of assurance claims useful in assuring resiliency maturity throughout industry, government, and defense.

The challenge associated with assuring the resiliency of systems of systems —based on a broad definition for resiliency—calls for a framework for assuring resiliency under stress expressed in terms of management, process, and engineering indictors useful in asserting resiliency assurance claims, validating assurance arguments, and verifying assurance evidence. The targeted users for assuring resiliency under stress include selected sectors within the critical infrastructure and defense industrial base and certain operating commands within the defense establishment. These are characterized by their increasing dependence on the acquisition, development, fielding, and sustainment of large-scale, complex systems of systems.

References

1. O'Neill, Don. "Preparing the Ground for Next Generation Software Engineering." Annual Technology Report. IEEE Reliability Society, 2008: 148-151.
2. "Software 2015: A National Software Strategy to Ensure U.S. Security and Competitiveness." Center for National Software Studies. 25 Apr. 2005 <www.cnsoftware.org/nss2report/NSS2FinalReport04-29-05PDF.pdf>.
3. "Future Directions in Software Engineering." *Software Tech News* 10.3. Oct. 2007 <www.softwaretechnews.com/pdf/stn10_3.pdf>.
4. O'Neill, Don. "Calculating Security Return on Investment." *Build Security In*. DHS National Cybersecurity Division. 6 Feb 2007 <https://buildsecurityin.us-cert.gov/daisy/bsi/articles/knowledge/business/677-BSI.html>.
5. "Critical Thinking: Moving From Infrastructure Protection to Infrastructure Resilience." George Mason University School of Law: Critical Infrastructure Protection Program. Discussion Paper Series. Feb. 2007 <http://cip.gmu.edu/archive/CIPP_Resilience_Series_Monograph.pdf>.

Table 2: *Framework for Assuring Resiliency Under Stress Goals and Indicators*

Framework for Assuring Resiliency Under Stress*	Focused Management Review	Defined Process Capability	Designed Engineering Solution
Demonstrate commitment to security assurance through strategic management, internal processes, and defense-in-depth.	• Competitiveness vs. security assessment and tradeoff. • Security return on investment. • Incident management.	• Chief Security Officer leadership program. • Security assurance operations. • Configuration management.	• Encryption. • Identity management. • Access control. • Authorization management. • Accountability management.
Demonstrate business continuity assurance through compliance management, external processes, and product engineering.	• Regulatory compliance. • Aspect oversight and assessment.	• Global sourcing. • Risk management. • Crisis management.	• Open source. • COTS software. • Security assurance evaluation tools.
Demonstrate the achievement of system survivability through the management of faults and failures, sustainability processes, and RMA engineering.	• Incident management. • Cyber forensics. • Management of defects, faults, and failures.	• Resistance. • Recognition. • Recovery. • Reconstitution.	• Reliabilty engineering. • Availability engineering. • Maintainability engineering.
Demonstrate the achievement of system of systems resiliency through the management of external interactions and dependencies, the control of distributed supervisory processes, and the practice of next generation software engineering.	• Coordinated recovery time objectives. • Interoperability of data and information exchange.	• Operation sensing and monitoring. • Digital situation awareness. • Distributed supervisory control. • Information and data recovery.	• Next-generation software engineering.

** The overall goal is to drive the business case and enterprise commitment towards the assurance of software security, business continuity, system survivability, and system of systems resiliency.*

6. Miller, Robert A., and Irving Lachow. "Strategic Fragility: Infrastructure Protection and National Security in the Information Age." *Defense Horizons* 59. Jan. 2008 <www.ndu.edu/ctnsp/defense_horizons/DH59.pdf>.
7. Goodenough, John, Howard Lipson, and Chuck Weinstock. "Arguing Security – Creating Security Assurance Cases." *Build Security In*. DHS National Cybersecurity Division. 4 Jan. 2007 <https://buildsecurityin.us-cert.gov/daisy/bsi/articles/knowledge/assurance/643-BSI.html>.
8. CMMI Product Team. "CMMI for Development, Version 1.2." SEI, Carnegie Mellon University. Technical Report CMU/SEI-2006-TR-008. Aug. 2006 <www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf>.
9. Caralli, Richard A., et al. "Introducing the CERT Resiliency Engineering Framework: Improving the Security and Sustainability Processes." SEI, Carnegie Mellon University. Technical Report CMU/SEI-2007-TR-009. May 2007 <www.sei.cmu.edu/pub/documents/07.reports/07tr009.pdf>.
10. Jackson, Daniel, Martyn Thomas, and Lynette I. Millett. *Software for Dependable Systems: Sufficient Evidence?* Washington, D.C.: National Academies Press, 2007.
11. Borg, Scott. "Recommendations for NDU Cyber Risk and Response Conference." U.S. Cyber Consequences Unit, National Defense University. Jan. 2009 <www.ndu.edu/CTNSP/cyberworkshop/1030%20BORG.pdf>.
12. Chess, Brian, Gary McGraw, and Sammy Migues. *The Building Security In Maturity Model*. 2009 <www.bsimm.com>.
13. "Fundamental Practices for Secure Software Development: A Guide to the Most Effective Secure Development Practices in Use Today." *SAFE Code*. 8 Oct. 2008 <www.safecode.org/publications/SAFECode_Dev_Practices1008.pdf>.
14. Collins, Rosann W., et al. "The CERT Function Extraction Experiment: Quantifying FX Impact on Software Comprehension and Verification." SEI, Carnegie Mellon University. Technical Note CMU/SEI-2005-TN-047. Dec. 2005 <www.sei.cmu.edu/pub/documents/05.reports/pdf/05tn047.pdf>.
15. O'Neill, Don. *Inside Track to Offshore Outsourcing Using the Trusted Pipe™: What Global Enterprises Look For in Offshore Outsourcing*. Proc. of the Making the Business Case for Software Assurance Workshop, Carnegie Mellon University, Pittsburgh. 26 Sept. 2008 <www.sei.cmu.edu/community/BCW_Proceedings.pdf>.
16. Ellison, R.J., et al. "Survivable Network System: An Emerging Discipline." SEI, Carnegie Mellon University. Technical Report CMU/SEI-97-TR-013. Nov. 1997, Rev. May 1999 <www.cert.org/research/97tr013.pdf>.
17. O'Neill, Don. "Maturity Framework for Assuring Resiliency Under Stress." *Build Security In*. DHS National Cybersecurity Division. 11 July 2008 <https://buildsecurityin.us-cert.gov/daisy/bsi/articles/knowledge/business/1016-BSI.html>.
18. Northrop, Linda. *Architecting High Quality Software: The Role of Software Architecture in System Development and Evolution*. Proc. of the 2nd Annual Team Software Process Symposium, Orlando, FL. Keynote Presentation. 19 Sept. 2007 <www.sei.cmu.edu/ tsp/symposium/2007/Day%203%20830AM%20SEI%20keynote.pdf>.

About the Author

Don O'Neill is a seasoned software engineering manager and technologist. As an independent consultant, O'Neill conducts defined programs for managing strategic software spanning competitiveness, security, and process improvement. Following his 27-year career with IBM's Federal Systems Division, O'Neill completed a three-year residency at the SEI under IBM's Technical Academic Career Program, and currently serves as an SEI visiting scientist. He served on the executive board of the IEEE Software Engineering Technical Committee and as a distinguished visitor of the IEEE. He is a founding member of the Software Process Improvement Network and served as the president of the Center for National Software Studies.

9305 Kobe WY
Montgomery Village, MD 20886
Phone: (301) 990-0377
E-mail: oneilldon@aol.com