A WHITE PAPER

"IA FUSION" Fusing IA into the Software Development Model for a Department of Defense (DoD) Contractor

Topic Summary

Fusing Information Assurance into the Software Development process reduces costs and increases quality because:

- Scarce, expensive, and overburdened IA personnel are no longer the sole resources available to ensure a project's IA compliance.
- Development team members can ground their decisions and recommendations in sound security practices throughout system design and implementation.
- Lower rework rates translate to lower project costs overall.
- Organizational capabilities improve and provide the contractor with a competitive edge in the DoD marketplace.

The tools and techniques presented in this paper can be incorporated into both Agile and Waterfall project management models throughout the DoD.



Executive Summary

Recent headlines serve as a sobering reminder of the consequences facing organizations that fail to ensure the confidentiality, integrity, and availability of their information systems. Nowhere are these perils more pronounced than within projects created for and delivered to the Department of Defense (DoD):

- A front-page story in the Wall Street Journal detailed how a hacker stole personal data on 77 million people from an online gaming network.¹ Considering the movement afoot in DoD to implement social networking within military sites such as Army Knowledge Online (AKO), this type of attack can be expected to recur frequently.
- The SecureID token used by many within DoD to logon to protected military Web sites was hacked.² As of this writing, the ramifications of this breach have yet even to be fully understood (much less remediated).
- A DoD contractor associated with a security breach may be "exiled" from future business opportunities.³ Contractors must think more than twice about the consequences of a lax IA program.

Cybersecurity measures to address these and other issues come at greater and greater expense (budgeted at \$3.2 billion for fiscal 2010⁴), even as DoD and the federal government are being pushed hard to cut costs by U.S. Chief Information Officer (CIO) Vivek Kundra. His call to "turnaround or terminate at least one-third of poorly performing projects in their portfolio within the next 18 months"⁵ speaks tellingly to the cost and expense involved in cybersecurity certifications: a contractor must deliver secured systems expeditiously or risk being tagged as a "poor performer."

This paper provides a roadmap called "IA Fusion" for a DoD contractor seeking to cut costs even while improving IA. As Professor John Savage of Brown University testified to Congress, "it is better to build in security rather than try to add it after the fact" and "hardware and software vendors and network providers should be required to conform to reasonable cyber security guidelines."⁶ Decisions by individual contributors are the linchpin of IA; IA Fusion helps an organization to translate IA policies into IA practices at the ground level.

IA Fusion gives an organization the competitive edge necessary to survive and thrive in today's demanding DoD business environment by integrating IA throughout the software development process. Moreover, IA Fusion's holistic management, education, and measurement approaches can be extended to projects across the organization – even those projects not directly related to the DoD.

Ultimately, however, IA Fusion has as its goal the improvement of delivered information systems on behalf of the most important customer of all: the Warfighter.

¹ Nick Wingfield, Ian Sherr, and Ben Worthen. "Hacker Raids Sony Videogame Network," *WSJ.com*, April 27, 2011. <u>http://tinyurl.com/wsj-sony-hack</u> (accessed: April 27, 2011).

² Aliya Sternstein. "RSA security breach compromised federal network ID tool," *nextgov*, March 21, 2011. <u>http://www.nextgov.com/nextgov/ng_20110321_6511.php</u> (accessed: April 27, 2011).

³ Chandler Harris. "Security Breaches By Contractors Could Lead to DoD Exile," *ClearanceJobs.com*, January 7, 2011. <u>http://tinyurl.com/dod-exile</u> (accessed: April 27, 2011).

⁴ Aliya Sternstein. "Pentagon seeks \$3.2 billion for revised cyber budget," *nextgov*, March 24, 2011. <u>http://www.nextgov.com/nextgov/ng_20110324_2474.php</u> (accessed: April 27, 2011).⁴

⁵ Vivek Kundra, "25 Point Implementation Plan To Reform Federal Information Technology Management," December 9, 2010. <u>http://tinyurl.com/kundra-25</u> (accessed: April 27, 2011).

⁶ "John Savage Testifies Before Senate Committee on Cyber Security," *Brown University*, April 21, 2011. <u>http://sca2002.cs.brown.edu/news/2011/0421.jestestimony.html</u> (accessed: April 27, 2011).



Table of Contents

EXECUT	IVE SUMMARY	I
1.0	INTRODUCTION	1
1.1	Problems in the Current Model	2
1.2	"IA Fusion" Introduced	4
1.3	TOPICS FOR THIS PAPER	6
2.0	DOD POLICY GUIDANCE	6
2.1	Management View	7
2.1	.1 Waterfall Model	7
2.1		
2.1	•	
2.1		
2.2	IA CONTROLS DURING THE DEVELOPMENT PHASE	11
2.2	2.1 DIACAP Knowledge Service (KS)	
2.2		
2.2	2.3 IA Control Selection	
2.3	IA CONTROLS AND EXPECTED PROJECT ARTIFACTS	
2.3		
2.3	3.2 DIACAP Artifacts	
2.3	3.3 Summary	
3.0	IA FUSION IMPLEMENTATION	19
3.1	Management Support	19
3.1	.1 Planning for IA Fusion	
3.1	.2 Delivering the IA Message	21
3.2	INTEGRATING IA FUSION	22
3.2	2.1 Training and Resources (Education)	23
3.2	2.2 Development Tools and Techniques	26
3.2	2.3 Measurement	
4.0	CONCLUDING REMARKS	37
APPEND	DIX A: RECOMMENDATIONS BY COST AND PRIORITY	
A.1: S	UMMARY	40
A.2: F	ecommendations by Cost (Highest to Lowest)	40
A.2	2.1: Automated Artifact Generation	
A.2	2.2: IMAF Implementation	
A.2	2.3: Project Championship	50
A.2	2.4: Secure Coding	52
A.2	2.5: IA Control Awareness	55
	ECOMMENDATIONS BY PRIORITY	
A.4: S	ELECTED EMPLOYEE COSTS	57
APPEND	DIX B: MIL-STD-498, ISO/IEC 12207, AND IEEE/EIA 12207: WHAT IS THE STANDARD?	58
APPEND	DIX C: ACRONYMS AND ABBREVIATIONS	60



ABOUT THE AUTHOR	62
REFERENCE LIST	63
MISCELLANEOUS REFERENCES	65

Illustration Index

Figure 1: Systems Engineering Model from Milestone B to Milestone C	2
Figure 2: IA Fusion Model	4
Figure 3: Waterfall Software Development	7
Figure 4: Incremental Software Development Paradigm (Defense Acquisition University Learning Portal)	8
Figure 5: Spiral Model Paradigm (Defense Acquisition University Learning Portal)	8
Figure 6: Agile Scrum	
Figure 7: MAC Levels and Data Classification within DIACAP	. 12
Figure 8: DIDs representing historical software artifacts	. 13
Figure 9: Software Test Report DID	. 14
Figure 10: Example Instructions from STR DID	. 15
Figure 11: DIACAP Life-cycle	. 17
Figure 12: Example DIACAP Artifact	. 18
Figure 13: IA Fusion Management	. 19
Figure 14: IA Fusion integration phases	. 22
Figure 15: Education aspect of IA Fusion	. 23
Figure 16: DoD Instruction 8500.02 IA Control Subject Areas	. 23
Figure 17: NIST SP 800-53 Security Control Classes	. 24
Figure 18: IEEE/EIA 12207 mapping to MIL-STD-498 DIDs	. 24
Figure 19: Development Tools aspect of IA Fusion	. 26
Figure 20: Example CDRL for database schema	
Figure 21: Example Unit Test CDRL	. 29
Figure 22: Automated unit test integration within Visual Studio	
Figure 23: Secure Memory and C# (Source: MSDN)	
Figure 24: IMAF MOD Assessment including qualitative data	. 33
Figure 25: Bayesian Belief Network showing quantitative analysis	. 34
Figure 26: Example of a MOD Protocol	
Figure 27: SRR Phases and Activities	
Figure 28: Multiple event failures collude to cause failures	
Figure 29: Strong-Matrixed Organizational Outline	. 40
Figure 30: Example IBM DOORS Project Management Integration	. 42
Figure 31: Example Integrated Software Testing Tool with Customizable Output	. 43
Figure 32: Example C# Documentation Tags	. 44
Figure 33: The IMAF Qualitative Assessment	. 47
Figure 34: Notional Project Championship Process	. 50
Figure 35: Microsoft Secure Development Life-cycle	. 53
Figure 36: The Pedigree of IEEE/EIA 12207	. 58



Tables Index

Table 1: Selected Publications Providing IA Guidance	3
Table 2: DIACAP Package Contents	17
Table 3: NIST Software IA Training Resources	
Table 4: Summary of Cost Estimates	
Table 5: Estimated Costs for Artifact Generation	45
Table 6: IMAF Implementation Estimates	47
Table 7: Project Championship Costs and Timeline	51
Table 8: Secure Coding Cost Estimates	54
Table 9: IA Control Awareness	55
Table 10: Recommendations by Priority	56
Table 11: Selected Employee Costs for a 2,000-person Company	57



1.0 Introduction

Effective information assurance (IA) requires more than the piecemeal deployment of tools and techniques such as firewalls and penetration testing. Guidance from the Defense Acquisition Guide (DAG) clearly states that "information assurance requirements shall be identified and included in the design, acquisition, installation, operation, upgrade or replacement of all DoD information systems" (p 628). As such, IA should be a holistic strengthening of an information system's security posture throughout the development life-cycle, aligned with the following principles:

- *Confidentiality.* Information within the system can be accessed only by authorized entities. *Example:* Transmit network data using encryption.
- *Control.* The contractor and DoD secure the system both physically and logically. *Example:* Maintain the system and its related data in a secure facility.
- Integrity. The system reliably stores information throughout the information life-cycle. *Example:* Apply a one-way cryptographic function to data⁷ or write historical data to read-only media.
- *Authenticity.* Transmitted data is associated authoritatively with its sender. *Example:* Apply a digital signature to data sent via email.⁸
- *Availability.* Information is readily available to authorized entities when they need it. *Example:* Use redundant network connections for critical systems such that failure of one connection does not prevent access to the system.
- *Utility.* Authorized entities can process ("use") required information. *Example:* Ensure that required data decryption keys are stored securely; otherwise, encrypted data would be *available* without being *useful.*

The DoD requires each contractor to prove a delivered system's IA compliance by means of demonstration and inspection. *Demonstration* means that the contractor shows the system performing its functions; for example, by executing a business process workflow. *Inspection* means that the contractor ensures transparency into the logical functions and processes making up the system; for example, by performing software code reviews or by computing and reporting the complexity of a delivered software package. A DoD contractor deliver artifacts, to prove both demonstration and inspection, throughout a project's life-cycle.

⁷ A one-way cryptographic function is a mathematical function that is significantly easier to compute in one direction (the forward direction) than in the opposite direction (the inverse direction) and is commonly referred to as a "hash function." (RSA Laboratories, "What is a one-way function?" <u>http://www.rsa.com/rsalabs/node.asp?id=2188</u>, accessed April 22, 2011.) For integrity, a hash function is applied to the contents of a document to generate a unique "hash value"; any change to the document results in a different hash value. Approved algorithms such as the Secure Hash Algorithm (SHA) can be used to generate a hash value of a particular length; generally, the greater the length, the more assurance the data owner has that the data cannot be modified without detection. As of October 14, 2011, the DoD is moving toward SHA-256, which creates a 256-bit hash value (<u>http://tinyurl.com/dodp-sha256</u>, accessed May 25, 2011).

⁸ Signing is a two-step process that uses public-key cryptography. First, a hash value is calculated for data. The hash value is then encrypted with the data owner's private key to create a digital data signature. Because only the data owner's public key can decrypt the encrypted hash value, the data owner cannot deny creating the digital signature. As with all public-key cryptography, it is imperative that the data owner maintain control of the private key. Chapter 7 from the Computer Security Handbook (BOS09 in Reference List) provides further information about public-key cryptography.



1.1 Problems in the Current Model

Software development for an information system normally occurs during the Engineering and Manufacturing Development (EMD) phase of a project's acquisition life-cycle. The DAG lays out an expected flow from Milestone B to Milestone C.⁹

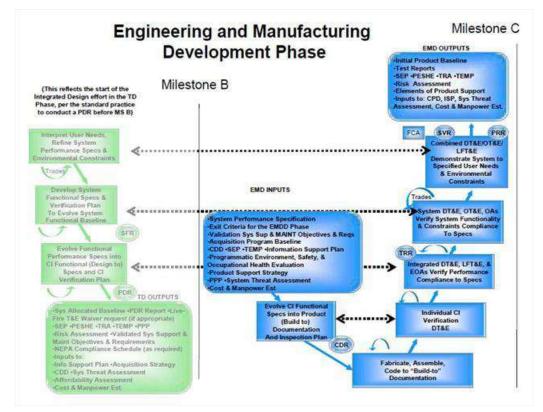


Figure 1: Systems Engineering Model from Milestone B to Milestone C^{10}

The DoD provides guidance for infusing any system's engineering life-cycle with IA. This guidance comes from both DoD publications and National Institute of Standards and Technology (NIST) Special Publications (SPs), just a few of which are outlined in the following table:

⁹ The DAG identifies three major milestones of the acquisition life-cycle: by Milestone A, the government has identified why a project should be pursued; by Milestone B, a contractor has been selected as a result of a Request for Proposal (RFP) or Request for Quote (RFQ); by Milestone C, the contractor has delivered the completed work product to the government for acceptance testing. See DAG Chapter 4 in the Reference List for more information.

¹⁰ Source: DAG, p 238.



Publication	Purpose
DoD 8500.01	Information Assurance; defines IA policy and objectives
DoD 8500.02	Information Assurance Implementation; IA controls and specifications based on system categorization
DoD 8510.01	DoD IA Certification and Accreditation (DIACAP); road map for system certification
DoD 8570.01-M	Workforce Improvement Plan; security training and certification for the DoD and contractor IT workforce.
NIST SP 800-50	Building an Information Technology Security Awareness and Training Program
NIST SP 800-53	Recommended Security Controls for Federal Information Systems; SP 800-53 defines 205 controls among 18 categories and is more granular than DoD 8500.02, which defines 157 controls among 8 categories its 157 controls and 8 categories are more granular than DoD 8500.02,
NIST SP 800-64	Security Considerations in the Systems Development Life-cycle

These publications depict IA as being layered on top of existing processes rather than fused into those processes:

DoD 8500.01: "All DoD information systems shall be certified and accredited..." (DOD-8500.1, p 5).

DoD 8500.02: "DoD Component-level IA programs shall include a standard convention for naming and describing IA functions; tracking their association with positions, roles, and contracts; and tracking the training and certification of personnel assigned to the positions, roles or contracts" (DOD-8500.2, p 37).

DoD 8510.01: "The System Manager shall...[e]nsure annual reviews of assigned ISs required by FISMA are conducted" (DOD-8510.1, p 8).

DoD 8570.01-M: "Provides guidance for the identification and categorization of positions and certification of personnel conducting Information Assurance (IA) functions within the DoD workforce supporting the DoD Global Information Grid (GIG)" (DOD-8570.1, p 11).

NIST SP 800-50: "The IT Security Manager has tactical-level responsibility for...ensur[ing] that awareness and training material is effectively deployed to reach the intended audience" (NIST 800-50, p 15).

NIST SP 800-53: "Assurance can be obtained in a variety of ways including...[a]ctions taken by security control assessors to determine the extent to which the controls are implemented correctly" (NIST 800-53, p 25).

NIST SP 800-64: "Security control settings are enabled in accordance with manufacturers' instructions, available security implementation guidance, and documented security specification...ISSO [*Information System Security Officer*] should review installed system to ensure that controls are in place and properly configured" (NIST 800-64, p 36).

While such an "IA layering" approach does help to ensure the quality control of a delivered information system, this paper contends that it insufficiently empowers other contributors to the system's development. By engaging and energizing software developers during the system's build phase, the DoD contractor can improve IA and minimize the time necessary to deliver and certify a secured system.

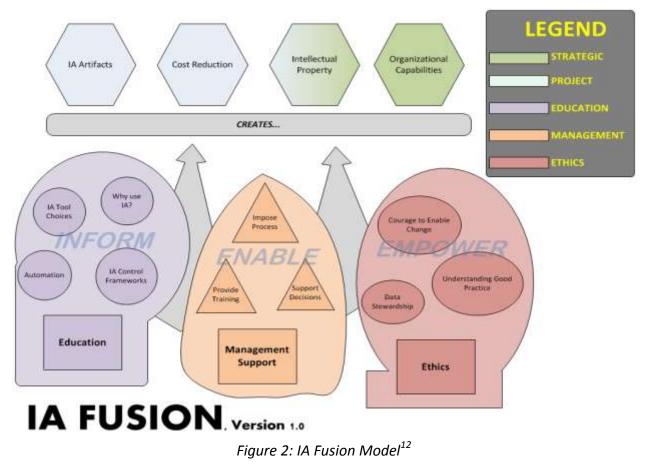
¹¹ See the Reference List for these works.



1.2 "IA Fusion" Introduced

IA Fusion is an organic methodology for integrating IA into the software development process. IA should not be compartmentalized as *inspection* or *demonstration* adjunctive to the actual development of software; rather, software developers (the individual contributors) themselves can serve as the contractor's primary IA resource. As the Information Assurance Technology Analysis Center (IATAC) put it in a 2007 State-of-the-Art Report (SOAR), "there is growing consensus within the software security assurance community that all participants in a software development project (including managers) need at least some knowledge of security" (p 298).

The proactive IA Fusion approach can be visualized as a set of mutually-reinforcing structures and concepts:



¹² Drawing by the author.



IA Fusion's foundational pillars include:

- Education. The stereotypical software developer is driven by the joy of technology for its own sake and regards IA as of secondary importance.¹³ Yet the prudent developer grounds all implementation decisions in IA best practices. By ensuring that software developers are aware of requirements from DoD documentation such as Security Technical Implementation Guides (STIGs) and IA controls from DoD Instruction 8500.2 ("Information Assurance (IA) Implementation"), Project Managers (PMs) can leverage the developers' skill and judgment while applying IA controls to a specific project.
- 2. *Support.* The manager needs to champion the developer within the organization and support the developer's IA-aware decisions. Such an IA-aware decision might be to propose automation techniques for generating required work product documentation: the PM must support such decisions by negotiating with decision makers to procure time and resources for the effort.
- 3. *Measurement*. Once implemented, IA Fusion tools and techniques must be continuously monitored and improved. Baselines of existing project metrics must be captured (such as system downtime due to software errors) and compared to the actual results for individual IA Fusion initiatives. This allows the PM to quantify return on investment (ROI) to the DoD customer and to the contractor's own management structure.

The IA Fusion approach ensures that software developers become the contractor's most valuable and effective IA assets by empowering them to analyze and report on IA implications during the system build or integration phase. IA Fusion yields four specific benefits at the project (system) and organizational (strategic) levels:

- *IA Artifacts.* Individual contributors are best situated to identify additional artifact automation and generation opportunities. This might include building internal design documents that can be readily translated to official contract-specified artifacts by using low-cost and reliable tools.
- *Cost Reduction.* Individual contributors make IA-aware decisions during the development phase, preventing IA from being compromised down the road and thereby saving the cost of correcting such failures. Consider confidential password data processed as part of a distributed transaction: although this data is encrypted when passed over network transmission media, the IA-minded developer would be aware that a weak point exists where the password is decrypted in memory while being received by the application. The educated software developer can ensure that such data is processed using "secure" memory, thus eliminating the need for possible rework during the system's certification and accreditation (C&A) process.
- Shared Intellectual Property. IA-based development processes created by individual contributors to one project can be applied to other projects throughout the organization. This expansion of the IA toolset

¹³ Infosecurity, "Insecure software plays key role in creating cybersecurity vulnerabilities," *infosecurity.com*, February 18, 2011. <u>http://www.infosecurity-magazine.com/view/16066/insecure-software-plays-key-role-in-creating-cybersecurity-vulnerabilities/</u> (accessed: April 25, 2011).



lowers costs and fosters collaboration between all of the DoD contractor's software development teams.

• *Organizational Capabilities.* The expertise engendered through the adoption of IA Fusion can be honed into a core competency, lending a competitive advantage to the DoD contractor.

1.3 Topics for this Paper

This paper approaches IA Fusion by starting at the top (high-level) and then working down to practical implementation steps:

- DoD Policy. Policy directs and informs all DoD contract implementations. This paper reviews project
 management techniques used within the DoD to build information systems, how IA controls are applied
 to such projects, and the artifacts typically generated from the development process. A thorough
 understanding of the DoD project landscape ensures that solutions are acceptable to the contractor's
 DoD customer.
- *Implementation.* This paper presents specific tools and techniques for implementing IA Fusion within software development. Management should develop marketing techniques for educating software developers about IA, and budget for education and implementation initiatives. Measurement is essential to ensure a proper ROI.

The paper closes with a summary of the IA Fusion approach and Appendices containing cost and schedule estimates for implementation.

2.0 DoD Policy Guidance

Policy from DoD authorities sets the IA requirements to which a project must adhere:

- DoD Instruction 5000.02 ("Operation of the Defense Acquisition System"): "Conduct information assurance testing on any system that collects, stores, transmits, or processes unclassified or classified information" (DOD-5000.2, p 52).
- DoD Directive 8000.01 ("Management of the Department of Defense Information Enterprise"): "Information shall be considered a strategic asset to the Department of Defense; it shall be appropriately secured, shared, and made available throughout the information life-cycle to any DoD user or mission partner to the maximum extent allowed by law and DoD policy" (DOD-8000.1, p 2).
- DoD Instruction 8580.1 ("Information Assurance (IA) in the Defense Acquisition System"): "IA shall be implemented in all system and services acquisitions at levels appropriate to the system characteristics and requirements throughout the entire life-cycle of the acquisition" (DOD-8580.1, p 2).

This section lays out operational constraints imposed upon the software development team from a management view, an IA control view, and an artifact view.

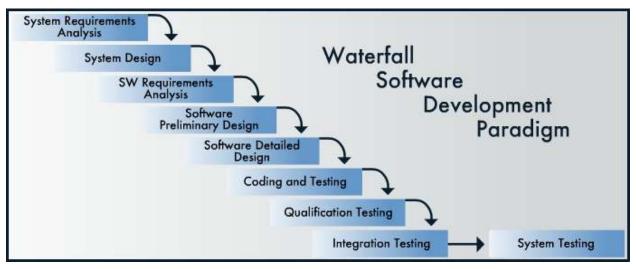


2.1 Management View

This section briefly explores both the Waterfall and Agile project management models.

2.1.1 Waterfall Model

The Waterfall model was formally described in the 1970s as a way of ensuring the quality of delivered work products via conformance to requirements while also mitigating risk.¹⁴ Waterfall emphasizes up-front planning and envisions the software development process as a sequential "relay race" where outputs from one process (such as Qualification Testing) are fed as inputs to another process (such as Integration Testing).



*Figure 3: Waterfall Software Development*¹⁵

The Waterfall approach is best-suited to systems whose requirements are well-defined and static. As the commercial software industry has matured and the development life-cycle has been compressed, private industry has largely abandoned the Waterfall method because of the difficulty in adjusting it to handle dynamic scheduling and capability requirements. Waterfall continues to be practiced in the DoD, where firm-fixed price (FFP) contracts abound and significant risk is shouldered by both the purchaser (DoD) and the vendor (DoD contractor). The standard DoD contract clearly specifies a project's scope, development plan, and required artifacts. Thus, Waterfall works well to ensure that the DoD customer receives exactly what it paid for.

¹⁴ The first formal description of the Waterfall model is often cited as an article published in 1970 by Winston W. Royce (1929 - 1995). Royce did not use the term "Waterfall" in this article, but instead presented the model as an example of a flawed methodology (ROYCE 70).

¹⁵ Source: Defense Acquisition University Learning Portal, <u>https://learn.dau.mil/CourseWare/43_4/scopage_dir/l5_acqstrat/l5_t3sdp.html</u> (accessed: April 25, 2011).



2.1.2 Variations on Waterfall

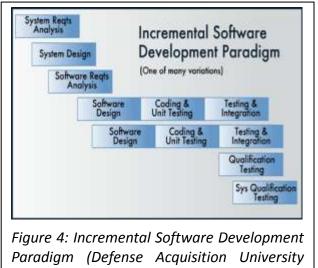
Significant cost and schedule overruns of large DoD contracts have often stemmed from the long delay between

the specification of a system's requirements and the actual delivery of that system.¹⁶ Since the 1970s, the DoD has implemented variations on Waterfall that attempt to shorten that interval. For example, an "incremental" development model breaks the design, coding, and testing of software into discrete sub-projects as shown to the side.

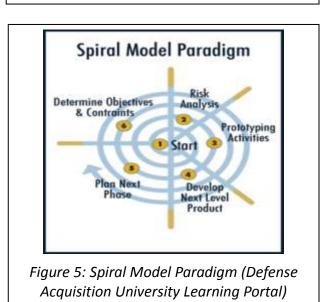
Under this model, requirements are still identified in advance of project execution. However, each delivery occurs as a smaller "chunk" with more frequent feedback to project sponsors. This allows for easier identification and correction of project problems than do traditional Waterfall techniques. Such an approach, however, is still not well-suited to handling evolving project requirements.

Another approach has been to break the software development process into "spirals" that provide evolutionary capabilities. Each spiral is delimited by an evaluation phase known as a "kill point"¹⁷ at which the overall project can be terminated if necessary. However, the size and complexity of individual spirals making up a delivered project can still hinder adaptation to changing requirements.

The Waterfall and Iterative techniques are highly structured and formalized approaches to project management. As requirements are being gathered, all stakeholders must be fully engaged so that the desired business results are fully understood. All project planning is performed prior to project execution. Where a system's requirements are hazy or undefined (e.g., "not knowing what you don't know"), Waterfall-based project management can be challenging to lead to success.



Learning Portal)



¹⁶ Associated Press, "GAO Highlights DoD Cost Overruns," March 13, 2009. <u>http://www.military.com/news/article/gao-highlights-dod-cost-overruns.html</u> (accessed: April 25, 2011).

¹⁷ "When phases are sequential, the close of a phase ends with some form of transfer or handoff of the work product produced as the phase deliverable. This phase end represents a natural point to reassess the effort underway and to change or terminate the project if necessary. These points are referred to as phase exits, milestones, phase gates, decision gates, stage gates or kill points" (PMI, p 19).



2.1.3 Agile

According to a December 2010 MITRE report created for the DoD, the Waterfall approach to software project management has been "misconstrued as a strictly sequential approach … reinforced by government contracting standards" (MITRE, p 18). In that same report, the essence of effective software development is described as being related to Dr. W. Edward Deming's "Plan-Do-Study-Act" model as a series of "time-boxed, iterative, and incremental product development plans." Dissatisfaction with sequential models of project management which emphasize artifacts and work processes has led to the development of Agile project management.

In 2001, Kent Beck and other software developers codified the "Agile Manifesto"¹⁸ as a reaction to traditional project management:

We are uncovering better ways of developing software by doing it and helping others
do it. Through this work we have come to value:Individuals and interactions over processes and toolsWorking software over comprehensive documentationCustomer collaboration over contract negotiationResponding to change over following a plan.That is, while there is value in the items on the right, we value the items on the left
more.

A 2010 Forrester Research study found that over half of its respondents had adopted Agile methods.¹⁹ As an evolving project management discipline, Agile has sometimes been dismissed by critics as an invitation to discard planning and structure and descend into chaos. Yet as the MITRE report recounts, discipline and expertise are the keys to effective implementation of Agile: "Agile methodologies require team members to be highly skilled and disciplined. Unlike traditional software development methodologies where each phase of the life-cycle sequentially informs the next, Agile methods perform these life-cycle phases essentially in parallel...[t]his requires team members to be intimately familiar with all phases of the software development life-cycle and allows little time for 'learning-on-the-job'" (MITRE, p 20).

Agile projects require close collaboration among end-users, sponsors (funding parties), and the software development team. Changes to requirements are not only expected but are actively encouraged; the assumption is that no one person or group fully understands the target audience's needs up-front.

¹⁸ Source: <u>http://agilemanifesto.org/</u> (accessed: April 25, 2011).

¹⁹ Mitchell Pronsc, "Agile is for Real, says Forrester," *AgileZone*, January 25, 2010. <u>http://agile.dzone.com/articles/45-</u> <u>developers-surveyed-use</u> (accessed: April 25, 2011).



According to the same Forrester Research study, the most popular Agile methodology is the "Scrum":

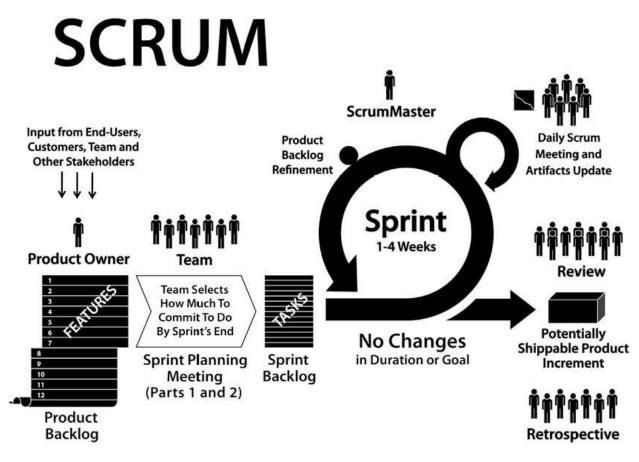


Figure 6: Agile Scrum²⁰

In Agile Scrum,²¹ a Product Owner works directly with the Team to define features; these features are stored in a product backlog (otherwise known as a "burn-down chart"). The Team works with the Product Owner to prioritize and select a set of features to execute within each cycle ("Sprint"). The result is a "potentially shippable product increment" that can be demonstrated to the Product Owner. Feedback from the Product Owner is applied to the product backlog to help influence future Sprints. Key characteristics of Scrum include:

• The Team is self-organizing; there is no project manager as such. Instead, a Scrum Master works to remove obstacles and to protect the Team. Daily Scrum Meetings, no longer than 15 minutes each, hone the team's focus. Team members report what they have accomplished during the past 24 hours,

²⁰ Deemer et. al., "The Scrum Primer version 1.2," *The Scrum Alliance*, 2009. <u>http://www.scrumalliance.org/resource_download/339</u> (accessed: April 25, 2011).

²¹ Ibid. The paragraph is paraphrased from the Primer.



what they propose to accomplish in the next 24 hours, and barriers to their forward movement that the Scrum Master can work to eliminate.²²

- Upon completion of a Sprint, the Team performs its own "retrospective" whereby continuous process improvement occurs.
- The success of the project is exemplified by "working code;" that is, software which is always potentially shippable to customers.
- Change is expected and valued as being the legitimate result of evolving understanding of business needs. Up-front planning is minimized.

For all the advantages of the Agile model, even its most ardent proponent will acknowledge that certain highrisk projects (such as medical equipment control systems) are not ideally suited to its use.²³

2.1.4 Summary

This paper describes how IA Fusion can be applied regardless of the method by which a particular project is being managed: in Waterfall, during the Coding and Unit Testing phases; in Agile Scrum, during individual Sprints. In either case, individual contributors must apply IA techniques and knowledge throughout the development process to ensure that the project meets the security and reliability requirements of the contractor's DoD customer.

2.2 IA Controls during the Development Phase

DoD policy guides which IA controls apply to a contractor's delivered products. These controls are defined by Enclosure 4 ("Baseline Information Assurance Levels") of Instruction 8500.2 ("Information Assurance (IA) Implementation"). To be accredited to run within a military environment, systems must adhere to DoD Instruction 8510.01 ("DoD Information Assurance Certification and Accreditation Process (DIACAP)").

2.2.1 DIACAP Knowledge Service (KS)

The DoD provides specific IA control guidance via the DIACAP Knowledge Service (KS).²⁴ This online portal hosts a collaboration tool where practitioners within the DoD IA community can exchange knowledge and stay notified of emerging security and accreditation trends.

2.2.2 DoD IA Controls and NIST

NIST issues guidance and direction to the Executive branch of the federal government. While NIST and DoD often work together to set standards and guidelines, NIST SP 800-53 ("Recommended Security Controls for Federal Information Systems and Organizations") defines a different set of security controls than does DoD's 8500.2. The Committee on National Security Systems (CNSS) directed DoD to integrate its 8500.2 IA controls with both SP 800-53 and NIST's general Risk Management Strategy in its October 2009 Instruction 1253

²² Traditionally, Scrum Meetings require all participants to remain standing in order to keep the meeting tightly focused.

²³ Robert Benjamin, "Know when to not use agile methodology," July 22, 2009, IT Toolbox, http://it.toolbox.com/wiki/index.php/Know_when_to_not_use_agile_methodology (accessed: May 15, 2011).

²⁴ The DIACAP KS is available at <u>https://diacap.iaportal.navy.mil/ks/</u>, but requires a Common Access Card (CAC) in order for the user to create an account and login.



("Security Categorization and Control Selection for National Security Systems").²⁵

As of this writing (May, 2011), the DIACAP KS cautions that "[t]he publication of those documents did not cancel or supersede the DIACAP. In order to prevent false starts and unnecessary enterprise-wide confusion during transition, the DIACAP will continue to be the effective DoD IA risk management process until a new DoD risk management process is published sometime this year."²⁶ Thus, the IA control integration between DoD and NIST should be considered as in-progress. Fortunately, the KS publishes numerous resources to aid a DoD contractor in standardizing on the NIST IA controls while still demonstrating compliance to the DoD IA controls defined within Instruction 8500.2; thus, the NIST SP 800-53 IA controls can be incorporated into a successful DIACAP implementation.

2.2.3 IA Control Selection

IA controls are tailored to each individual project. The current DIACAP model calls for the Program Management Office (PMO) to evaluate the project's Mission Assurance Category (MAC) based on the highest level of data that the project stores or processes, as shown below:

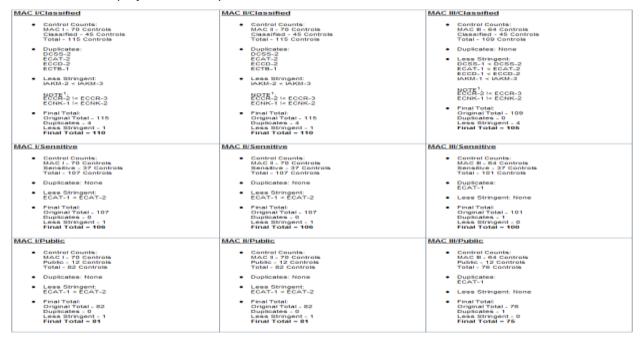


Figure 7: MAC Levels and Data Classification within DIACAP²⁷

²⁷ Source: DIACAP KS (CAC required),

https://diacap.iaportal.navy.mil/ks/ImplementationGuidance/Controls/Pages/ControlSetAnalysis.aspx (accessed: April 25, 2011).

²⁵ The reader can refer to the Instruction at <u>http://www.cnss.gov/Assets/pdf/CNSSI-1253.pdf</u>, especially page 9.

²⁶ DIACAP KS (CAC required), "Has the DIACAP been cancelled now that the new, transformational IA risk management policies (NIST SP 800-37, Revision 1; NIST SP 800-53, Revision 3; and CNSSI 1253) are published?" https://diacap.iaportal.navy.mil/ks/ImplementationGuidance/Transformation/Pages/default.aspx (accessed: April 25, 2011).



MAC levels range from I ("Systems handling information that is determined to be vital to the operational readiness or mission effectiveness of deployed and contingency forces") to II ("Systems handling information that is important to the support of deployed and contingency forces") to III ("Systems handling information that is necessary for the conduct of day-to-day business, but does not materially affect support to deployed or contingency forces in the short-term").²⁸ The end result is a set of IA controls that, in conjunction with a project's overall risk management strategy, cost-effectively secures a system based upon its possible impact to the nation.

2.3 IA Controls and Expected Project Artifacts

All Requests for Proposal (RFP) and Requests for Quote (RFQ) issued by a DoD Agency or Service ("Component") specify Contract Data Requirements Lists (CDRLs); or artifacts that must be delivered so that the DoD customer can be certain that the finished work product conforms to all contract requirements. These CDRL artifacts represent both due diligence (the government's research into identifying the artifacts) and due care (the government's rigor in receiving, verifying, and storing the artifacts).

2.3.1 ISO/IEC 12207 and MIL-STD-498

In 1996 the DoD standardized on International Organization of Standards / International Electrotechnical Commission (ISO/IEC) 12207 ("Information Technology: Software Life-cycle Processes") to replace the Military Standard (MIL-STD) 498 ("Software Development and Documentation"). Prior to the adoption of ISO/EIC 12207, DoD defined Data Item Descriptions (DIDs) within MIL-STD-498; these DIDs identified twenty-two different types of deliverables as shown below:

498-STD.PDF 498-IDX.PDF 498CBIDX.PDF	 Software Development and Documentation Full hypertext version of MIL-STD-498 inde Combined Index of major references in MII Guidebook, and the Overview & Tailoring 6 	ex (separate PDF fi L-STD-498, the MI	L-STD-498 DIDs, the Application & Reference
COM.DID.PDF	- Computer Operator Manual DID	SIP-DID.PDF	- Software Installation Plan DID
CPM-DID.PDF	- Computer Programming Manual DID	SPS-DID.PDF	 Software Product Specification DID
DBDD-DID.PDF	- Database Design Description DID	SRS-DID.PDF	- Software Requirements Specification DID
FSM-DID.PDF	- Firmware Support Manual DID	SSDD-DID.PDF	- System/Subsystem Design Description DID
IDD-DID.PDF	- Interface Design Description DID	SSS-DID.PDF	- System/Subsystem Specification DID
IRS-DID, PDF	- Interface Requirements Specification DID	STD-DID.PDF	- Software Test Description DID
OCD-DID.PDF	- Operational Concept Description DID	STP-DID.PDF	- Software Test Plan DID
SCOM-DID.PDF	- Software Center Operator Manual DID	STR-DID.PDF	- Software Test Report DID
SDD-DID.PDF	- Software Design Description DID	STRP-DID.PDF	- Software Transition Plan DID
SDP-DID.PDF	- Software Development Plan DID	SUM-DID.PDF	- Software User Manual DID
SIOM-DID.PDF	- Software Input/Output Manual DID	SVD-DID.PDF	- Software Version Description DID
has a second			

Figure 8: DIDs representing historical software artifacts²⁹

²⁸ DODI 8500.02, p 22.

²⁹ Source: "MIL-STD-498 PDF Roadmap" from Abelia Corporation, <u>http://www.abelia.com/498pdf/roadmap.pdf</u> (accessed: April 25, 2011).



While these DIDs no longer represent the DoD's mandatory artifact standards, contract officers (KOs) still draw upon them to determine the specific set of artifact deliverables required of a given system. As an example, consider the Form DD ("Department of Defense Form") 1664 ("Software Test Report" or STR), a portion of which is shown below:

	TA ITEM DESCRIPTION		Form Approved		
Public reporting burden for collection of this information is estimated to average 110 hours per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate of Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Ariington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington , DC 20503.					
1. TITLE			2. IDENTIFICATION NUMBER		
SOFTWARE TEST REPORT (STR)			DI-IPSC-81440		
3. DESCRIPTION/PURPOSE					
 3.1 The Software Test Report (STR) is a record of the <u>qualification testing</u> performed on a Computer Software Configuration Item (CSCI), a software system or subsystem, or other software-related item. 3.2 The STR enables the acquirer to assess the testing and its results. 					
4. APPROVAL DATE	5. OFFICE OF PRIMARY RESPONSIBILITY	6a. DTIC APPLICABLE	6b. GIDEP APPLICABLE		
941205	EC	APPLICADLE	AFFLICABLE		
7. APPLICATION/INTERREL	ATIONSHIP				
 7.1 This Data Item Description (DID) contains the format and content preparation instructions for the data product generated by specific and discrete task requirements as delineated in the contract. 7.2 This DID is used when the developer is tasked to analyze and record the results of CSCI qualification testing, system qualification testing of a software system, or other testing identified in the contract. 					
7.3 The Contract Data Requirements List (CDRL) (DD 1423) should specify whether deliverable data are to be delivered on paper or electronic media; are to be in a given electronic form (such as ASCII, CALS, or compatible with a specified word processor or other support software); may be delivered in developer format rather than in the format specified herein; and may reside in a computer-aided software engineering (CASE) or other automated tool rather than in the form of a traditional document.					
7.4 This DID supersede	7.4 This DID supersedes DI-MCCR-80017A, DI-IPSC-80698, and DI-MCCR-80311.				
	Figure 9: Software Test Repo	ort DID ³⁰			

³⁰ Source: Abelia Corporation, <u>http://www.abelia.com/498pdf/STR-DID.PDF</u> (accessed: April 25, 2011).



The artifact as displayed above does not much inform the KO as to software testing requirements, but the attached instructions starting from Section 3 specify a road map of what the KO should expect from the contractor:

 Overview of test results. This section shall be divided into the following paragraphs to provide an overview of test results.

- 3.1 Overall assessment of the software tested. This paragraph shall:
 - Provide an overall assessment of the software as demonstrated by the test results in this
 report
 - Identify any remaining deficiencies, limitations, or constraints that were detected by the testing performed. Problem/change reports may be used to provide deficiency information.
 - c. For each remaining deficiency, limitation, or constraint, describe:
 - Its impact on software and system performance, including identification of requirements not met
 - 2) The impact on software and system design to correct it
 - 3) A recommended solution/approach for correcting it

3.2 <u>Impact of test environment</u>. This paragraph shall provide an assessment of the manner in which the test environment may be different from the operational environment and the effect of this difference on the test results.

3.3 <u>Recommended improvements</u>. This paragraph shall provide any recommended improvements in the design, operation, or testing of the software tested. A discussion of each recommendation and its impact on the software may be provided. If no recommended improvements are provided, this paragraph shall state "None."

Figure 10: Example Instructions from STR DID³¹

The STR must include overall assessment of the software testing method, deficiencies, limitations, constraints, impact on other software packages or systems, and improvements. Furthermore, while this particular DID has not been standard issue since 1996, consider that a 2009 solicitation from the Navy used almost identical terms (emphasis added):³²

³¹ Ibid.

³² Source: SOLICITATION, OFFER AND AWARD from NAVAL SURFACE WARFARE CENTER CRANE DIV dated December 4, 2009 (p 78). <u>https://www.neco.navy.mil/synopsis_file/N0016410SNB07_att.pdf</u> (accessed: April 25, 2011).



3.7.1 Software Testing

The contractor shall conduct software testing, proofing each software build prior to release to the Government. The contractor shall provide a **Software Test Plan** (CDRL A015). The Software Test Plan shall describe the software test environment to be used for testing, identifying the test to be performed, and providing schedules for test activities. The contractor shall provide a **Software Test Description** (CDRL A016). The Software Test Description shall describe the test preparations, test cases, and test procedures used to perform qualification testing. The contractor shall provide a **Software Test Report** (CDRL A017) to document the results of the software qualification test. The contractor shall provide annual updates to the test plan and test description over the life of the contract, as required. Test Reports shall be provided for each software build.

The DID format is an excellent starting point for a DoD contractor to use in proving its IA compliance with contractual terms. For a brief history of the various software standards used by DoD, the interested reader may refer to "Appendix B: MIL-STD-498, ISO/IEC 12207, and IEEE/EIA 12207: What is the Standard?"

2.3.2 DIACAP Artifacts

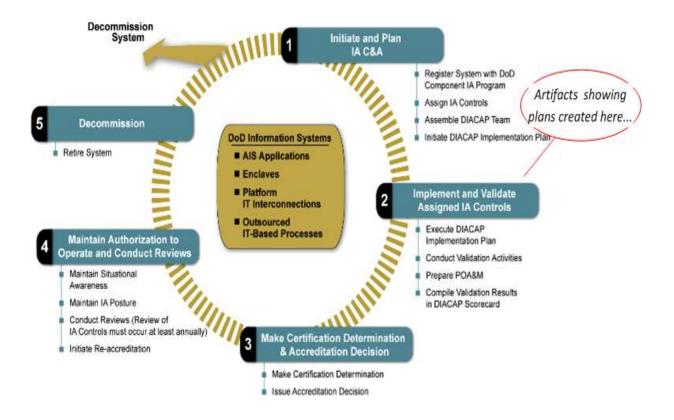
A delivered information system must run within a secure military environment; thus, the C&A provided by the DIACAP is always explicitly stated. The DoD's information systems are accredited by an authorized government representative to run for three year periods ("Authority to Operate," or ATO). Consider the following example from a September 2010 Performance Work Statement (PWS) issued by the U.S. Army:³³

DoD Information Assurance Certification and Accreditation Process (DIACAP) will be used on new efforts; system currently certified and accredited under DITSCAP will be transitioned to DIACAP.

DIACAP is performed at the end of Milestone B as part of the system delivery process. Thus, DIACAP artifacts must be considered ahead of time; especially, during the system build phase. Note that in the following diagram, artifacts are understood to be typically created during the IA control validation activity after the deliverable has been created. The IA Fusion approach posits that these artifacts should be planned for and generated during the system's development life-cycle.

³³ U. S. Army CECOM Life-cycle Management Command, Performance Work Statement (PWS) for the Rapid Response -Third Generation (R2- 3G) Program Revision 4, September 2, 2010, pg. 26. http://r23g.adamscomm.com/downloads/R2-3G%20Basic%20Contract%20PWS.doc (accessed: April 26, 2011).





*Figure 11: DIACAP Life-cycle*³⁴

The DoD does not mandate specific artifact templates but instead requires security categories to be addressed as part of a "DIACAP Package," which contains five specific elements. These elements are paraphrased in the following table:

Element	Description
System Identification Package (SIP)	Establishes the relationship between the DoD IS and the governing DoD Component IA program.
DIACAP Implementation Plan	IA controls (inherited / implemented); Implementation status; Responsible entities; Resources; Estimated completion date for each IA control.
Supporting CertificationActual validation results and the artifacts associated with impler of IA controls.	
DIACAP Scorecard	Summary report that succinctly conveys information on the IA posture of a

³⁴ Source: DIACAP Knowledge Service (CAC required). <u>https://diacap.iaportal.navy.mil/ks/ImplementationGuidance/Activities/Pages/default.aspx</u> (accessed: April 27, 2011).





	DoD information system.
Plan of Action and Milestones (POAM)	As required based on the above.

The bolded "Supporting Certification Documentation" element from the above table defines the systemprovided artifacts that inform the DIACAP evaluation. The following screenshot details an example DIACAP artifact that must be provided for C&A to be successful:

5.4 Functional Architecture for AIS Applications (DCFA-1)					
<insert sys<="" td=""><td colspan="5"><insert functionalty="" system=""></insert></td></insert>	<insert functionalty="" system=""></insert>				
5.4.1 Application Access Control					
<describe></describe>	-				
5.4.2 External Interfaces (ECTM-1, ECTM-2)					
<describe></describe>					
<insert diagram="" interface=""> The following table describes all external interfaces, the local and remote systems, and the information exchanged.</insert>					
Source	Destination	Classification	Protocol	Port	Direction

*Figure 12: Example DIACAP Artifact*³⁵

2.3.3 Summary

Both the older-style DIDs specified within MIL-STD-498, as well as the more free-form requirements exemplified by ISO/IEC 12207 and the DIACAP process, require artifacts to be delivered in order to demonstrate a system's IA compliance. IA Fusion advocates involving individual software developers early in the system's development process to ensure that these artifacts are available, accurate, and created cost-effectively.

³⁵ Source: i-Assure, "Security Design Document (SDD) Artifact Version 1.0.0," October 23, 2007, <u>http://www.i-assure.com/forums/Attachment31.aspx</u> (accessed: April 27, 2011). The i-Assure site provides free DIACAP artifacts for practitioners via their forum page <u>http://www.i-assure.com/forums/Topic336-4-1.aspx</u>.



3.0 IA Fusion Implementation

IA Fusion integrates Management, Education, and Ethical components to create a successful program.

- Management Support. Enables the software team to apply IA in meaningful and creative ways.
- *Education.* Informs individual contributors of the importance of IA from the everyday (for example, ensure that contract payment milestone requirements are met) to the innovative (for example, automate the production of IA artifacts).
- *Ethics.* Empowers each contributor within the software team to assume responsibility for ensuring that IA is implemented within the project to its best capability.

This section provides an implementation overview for these topics.

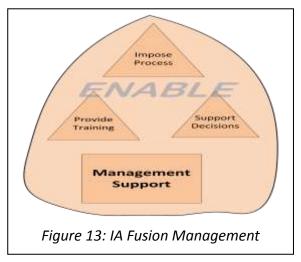
3.1 Management Support

3.1.1 Planning for IA Fusion

The importance of planning is such that planning processes constitute the majority of the 44 processes making up the Project Management Institute's Project Management Body of Knowledge (PMI, p 43). Three key management areas help this planning and assist with successful IA Fusion: *Process, Training,* and *Support*. This section explores each area and presents recommendations for the PM to consider.

The first and foremost aspect of IA Fusion planning is to get support and buy-in from senior management. While IA Fusion concentrates on empowering the individual software developers, only support from senior-level management can ensure that IA Fusion is aligned to corporate objectives and truly supports the DoD contractor's business requirements.

3.1.1.1 Process and Systems Engineering



Any successful venture, from developing a software system to constructing a road, must be grounded in a rigorous engineering process. The two management methodologies examined in this paper (Waterfall and Agile) differ as to how a project should be executed, but both agree on the need for solid engineering practices. Similarly, IA Fusion recognizes systems engineering as the single most critical project management tool.

The National Defense Industrial Association (NDIA) System Assurance Committee describes systems engineering from an IA stance (NDIA08, p 48-50), major points of which are listed here:³⁶

• Secure the hardware and software environments to be used by the developers. Demarcate "sandbox" environments in which developers can create new code without impacting the work done by other developers. Additionally, provide separate environments for code integration, quality assurance (QA),

³⁶ Engineering for System Assurance, http://www.acq.osd.mil/se/docs/SA-Guidebook-v1-Oct2008.pdf. In the Reference List this is located under NDIA08.



and test deployments. These environments should resemble the production environment to the greatest extent possible.

- *Require a software peer review.* In a Waterfall project, this would be performed prior to software delivery to QA. In an Agile Scrum project, this can be embedded into individual Sprints by using Pair Programming (SOAR, p 339).
- *Detect security failures.* The development team must be interviewed to determine what tools and techniques can detect IA problems early in the process. For example, the team can identify software tools such as automated code coverage tests and techniques like software complexity metrics analysis.
- Protect the Configuration Management (CM) environment. Ensure that software changes can be tracked to specific individuals and that when software modifications "break the build" the appropriate persons are notified as soon as possible.

3.1.1.2 Training

Software developers must integrate security engineering and assurance methodologies across the project lifecycle to avoid creating vulnerabilities. From the NDIA publication (NDIA08):

- Allocate funds for training programs. The training can be more or less formal depending on the sophistication of the developers. NDIA advocates creating an "assurance case" for the resources expended on IA to provide "justification to stakeholders that critical system assurance requirements are met" (p 16).
- As a special case for DoD programs, provide counterintelligence briefings for staff that must interact with external entities. For example, when developers work with commercial vendors, they must understand clearly what can and cannot be communicated.

3.1.1.3 Management Support

Software developers may feel that management and stakeholders take little interest in the purely technical problems that can plague an information system. The PM must counter this attitude by taking active measures:

- Create a communication path to the organization. Ensure that paths to the Change Advisory Board (or similar structure) are defined so that software developers will be confident of their ability to raise objections or volunteer information. This is not to say that developers should bypass their project manager (or Scrum Master/Product Owner in an Agile environment); consistent and successful change messages to upper management must originate from a single point of contact.
- Support the team members. When a team member has a valid concern over an issue that can affect the confidentiality, integrity, or availability of the system then the PM must actively stand behind that team member. It will not be unusual to receive pushback on such concerns from project stakeholders due to constraints imposed by the "iron triangle" of scope, costs, and schedule; however, for IA Fusion to work



the team members must have faith in their PM's "backbone" for delivering possibly unpopular (but important) messages.

As reported to the author by a project manager within a large DoD prime contractor, opening up these lines of communication requires a fundamental shift in how communications occur within existing projects. For "empowered change" communication to flourish, change control boards (CCBs) must be capable of receiving proactive change requests from staff. The PM must actively market said changes within the CCB, so the time required to do so must be budgeted as part of project planning.³⁷

3.1.2 Delivering the IA Message

Once appropriate management structures exist, the PM must plan a marketing strategy for delivering the IA message to the team. IA messaging should be:

- Focused. Start by examining the organization's security policies in conjunction with the specific project's charter. If the organization is compliant to ISO 9001:2008 ("Quality Management Systems") then all IA training must emphasize that tools and techniques must maintain conformance to that standard.³⁸ Thus, a version control system must exist in order to support the change documentation processes already put in place by the organization; this in turn may constrain the team's IA tool selection.
- Productive. Security guru Donn B. Parker notes that workers often believe that "they can work faster and better by not making backups, using pirated software, failing to securely store sensitive information" and following other bad practices (BOS09, p 1263). This attitude ignores the integrity and availability benefits reaped when IA has been sown: data backups and good configuration management help to provide a solid development infrastructure, while avoiding pirated or inappropriate software aids in keeping a well-defined development environment up and running reliably.
- *Relevant.* DoD's Information Assurance Support Environment (IASE) offers excellent free online IA training courses.³⁹ However, these courses demonstrate some of the difficulties inherent in educating software developers. As a case in point, the IASE's Computer Network Defense (CND) video (dated 2006) is geared toward "high-level managers."⁴⁰ Aside from being a bit out of date, the content lacks

³⁷ Source: Personal interview with Project Manager of a DoD prime contractor on May 6, 2011. Such changes must be considered in the light of project timelines and plans.

³⁸ Tools cannot be ISO 9001 compliant, only organizations. Tools must, however, support the formal processes that make the organization compliant. For an interesting blog article on the subject that explores this difference and also contains links to other ISO 9001 sites, the reader can see <u>http://stackoverflow.com/questions/3822825/what-are-theimplications-of-iso-9001-cmmi-for-source-control-in-general-and-git</u> (accessed: April 27, 2011).

³⁹ See IASE's full catalog at <u>http://iase.disa.mil/eta/online-catalog.html</u> (accessed: April 27, 2011). Some of the programs require a CAC login, but many are openly available to the public.

⁴⁰ See <u>http://iase.disa.mil/eta/cndv2/launchPage.htm</u> (accessed: April 27, 2011).



nuts-and-bolts instruction for software developers charged with securing networks. The PM must devise relevant and practical IA presentations for the development team.

3.2 Integrating IA Fusion

Each Agency and Service ("Component") within DoD must furnish "IA architecture and supporting master plan, coordination of IA projects across multiple investments, clear assignment of organizational roles and responsibilities, and development and management of a professional IA workforce" (DOD-8500.2, p 36). An information system's life-cycle must be imbued with IA from inception to disposal. This paper's IA Fusion fits squarely into the Implementation and Integration phases of the standard DoD V-Model, as shown in the following figure:

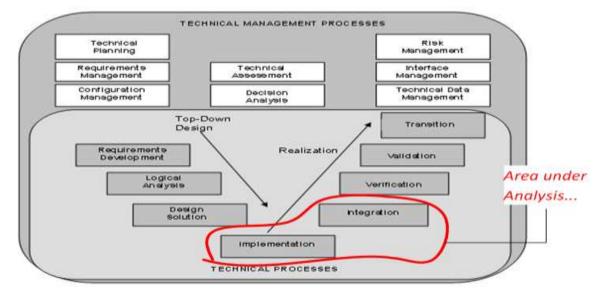


Figure 14: IA Fusion integration phases⁴¹

Individual contributors should be educated, empowered, and supported as per the following:

- *Training and Resources.* The development team must understand why IA is so important to a product's delivery.
- Development Tools and Techniques. The products used to develop systems must be capable of producing necessary IA artifacts and must contribute to the overall confidentiality, integrity, and availability of the delivered system while yielding a demonstrable cost-benefit.
- *Measuring Success.* "If you cannot measure it, you cannot improve it" (Lord Kelvin, 1883). A PM must present valid measurements to show the efficacy of the IA Fusion initiative.

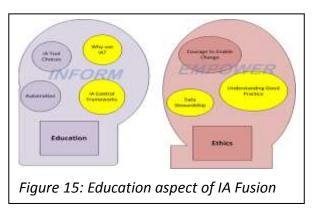
⁴¹ Drawing by the author. While the model implies a Waterfall project management environment, in an Agile Scrum environment IA Fusion would be targeted toward the individual Sprints.



3.2.1 Training and Resources (Education)

Both the DoD and NIST have developed a variety of training materials that lend themselves well to IA Fusion and help to ensure that software developers understand:

- why IA is so important;
- *what* IA should be held to; and,
- how day-to-day decisions can help (or hinder) not only security but also the generation of necessary IA artifacts.



3.2.1.1 DoD 8570.01-M – IA Workforce Improvement Plan

The DoD requires that the entire workforce (all authorized users) receive a minimum level of IA awareness training, and that the "IA technical workforce" performing "IA functions" must receive additional specialized training.⁴² These requirements mesh with IA Fusion, as experienced software developers can be educated on what IA must accomplish and can thus improve their ability to ground every decision in IA considerations. Unfortunately, the DoD relies primarily on industry certifications (DOD-8570.1, p 90) which are valuable but too general to assist the PM in assessing the suitability of a developer's IA qualifications for a particular project.

3.2.1.2 IA Controls

A better way for the PM to implement IA Fusion is to assign an IA resource to work directly with the system designers throughout the system life-cycle using deductive techniques like fault tree analysis and inductive techniques like failure mode and effect analysis (SOAR, p 64). A risk analysis should be performed to identify vulnerabilities within a given system's design and to analyze how IA controls can best mitigate exploitation of these vulnerabilities.

DoD requires the use of IA controls from Instruction 8500.02, which span eight "subject areas":

Abbreviation	Subject Area Name	Number of Controls in Subject Area
DC	Security Design & Configuration	31
IA	Identification and Authentication	9
EC	Enclave and Computing Environment	48
EB	Enclave Boundary Defense	8
PE	Physical and Environmental	27
PR	Personnel	7
CO	Continuity	24
М	Vulnerability and Incident Management	3

Figure 16: DoD Instruction 8500.02 IA Control Subject Areas⁴³

As mentioned earlier, DoD is moving towards a NIST-centric risk model based on the 205 IA controls defined by

⁴² IA functions "focus on the development, operation, management, and enforcement of security capabilities for systems and networks" (DOD-8570.1, p 16).

⁴³ Source: DOD-8500.2, p 49.



NIST SP 800-53⁴⁴ and the DIACAP site has already created a set of control mappings that correlate between NIST SP 800-53 and DoD 8500.02. Because the NIST controls offer more granularity and can be used to secure any information system to a level compatible with DoD, this paper standardizes on NIST SP 800-53 with its eighteen security control classes (CNSSI 1253 does not include the eighteenth control class "Program Management"):

DENTIFIER	FAMILY	CLASS
AC	Access Control	Technical
AT	Awareness and Training	Operational
AU	Audit and Accountability	Technical
CA	Security Assessment and Authorization	Management
CM	Configuration Management	Operational
CP	Contingency Planning	Operational
IA	Identification and Authentication	Technical
IR	Incident Response	Operational
MA	Maintenance	Operational
MP	Media Protection	Operational
PE	Physical and Environmental Protection	Operational
PL	Planning	Management
PS	Personnel Security	Operational
RA	Risk Assessment	Management
SA	System and Services Acquisition	Management
SC	System and Communications Protection	Technical
SI	System and Information Integrity Not included in CNSSI 1253	Operational
PM	Program Management	Management

Figure 17: NIST SP 800-53 Security Control Classes⁴⁵

3.2.1.3 IEEE/EIA 12207 and MIL-STD -498

As described above in the section "ISO/IEC 12207 and MIL-STD-498," the DoD deprecates the highlycustomizable, practical, and implementation-oriented DIDs that MIL-STD-498 made available. Fortunately, Mr. Michael S. Bandor of the Software Engineering Institute (SEI) within Carnegie Mellon University created a set of resources that map these venerable DIDs to their modern counterparts:

Information items(s)	References (See annex A.)	Data Item Description (DID) Template
Acceptance strategy and conditions record	IEEE 1062	
Acquisition plan	ASTM E731, E1206, IEEE 1062	
Acquisition requirements record	IEEE 1062, 1220	
Audit agenda record	2013 Sec. 2. 2010 Sec. 2010	
Change Request	an Contractory	2 00 00 00 00 10 10 00 00 00 00 00 00 00
Concept of operations description	IEEE 1362, EIA/IEEE J-STD-016 F.2.1. Also see ISO 5806, 5807, 8631, 8790, and 11411 for guidance on use of notations.	Operational Concept Description (OCD) DID DI-IPSC-81430A
Concept/need determination record	IEEE 1062, 1220	
Database design description	IEEE 1016, EIA/IEEE J-STD-016 G.2.3	Database Design Description (DBDD) DID DI-IPSC-81437A

Information Item Matrix (Table 1) from IEEE/EIA 12207 1 1997

Figure 18: IEEE/EIA 12207 mapping to MIL-STD-498 DIDs⁴⁶

⁴⁴ DIACAP guidance indicates "little or no impact" is expected from this transition, which is slated for 2011-2012.

⁴⁵ Source: NIST 800-53, p 17.



3.2.1.4 Other Training

IA training for software developers has been given short shrift. In fact, security maven Bruce Schneier deplores that "the IT security industry was born by accident, ignored all its life, and is now dying."⁴⁷ Commercial security credentials such as the Certified Information Systems Security Professional (CISSP)⁴⁸ or those available from the Global Information Assurance Certification (GIAC)⁴⁹, while extremely valuable, generally are not held by rank-and-file software developers. Fortunately, NIST provides security-centric training materials oriented towards the software developer:

Title	Location	Summary
SP 800-53 Rev. 3: Recommended Security Controls for Federal Information Systems and Organizations	http://csrc.nist.gov/pu blications/nistpubs/80 0-53-Rev3/sp800-53- rev3-final_updated- errata_05-01-2010.pdf	Chapter 3 provides a short and meaningful introduction to what IA controls look like and are designed to do. The software developers will understand and appreciate the multiple levels of controls matched to the system's categorization. The goal is for the developers to begin considering IA implications of their systems and functions.
SP 800-64 Rev. 2: Security Considerations in the System Development Life-cycle	http://csrc.nist.gov/pu blications/nistpubs/80 0-64-Rev2/SP 800-64- Revision2.pdf	This guide was written to be read from the CIO level down to system developers. Page 29 points out the need to consider C&A during the implementation process. The document's high-level scope and lack of specific strategies makes it more useful as an introductory guide.
SP 800-92: Guide to Computer Security Log Management	http://csrc.nist.gov/pu blications/nistpubs/80 0-92/SP 800-92.pdf	Of less practical use to software developers, this document is still useful because logging is the best way to troubleshoot problems and perform forensic incident analysis. A more software development-centric way to put logging is to use the phrase "code instrumentation;" the developers are more likely to understand that phrase.
SP 800-95: Guide to Secure Web Services	http://csrc.nist.gov/pu blications/nistpubs/80 0-95/SP 800-95.pdf	Highly informative and relevant to most software developers, this document describes the security pitfalls inherent in modern Web development. It covers how security can be applied using Security Access Markup Language (SAML). Unfortunately, it does not address the emerging perils presented by REpresentational State Transfer (REST) Web service calls.
SP 800-122: Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)	http://csrc.nist.gov/pu blications/nistpubs/80 0-122/sp800-122.pdf	Written for the organization, this paper covers the reasons for and importance of protecting personal data. Software developers within DoD must be cognizant of their duty to consider "what information is being stored, who is reading and writing that information, where is it going, and how is it sanitized."
SP 800-142: Practical Combinatorial Testing	http://csrc.nist.gov/gr oups/SNS/acts/docum	Heavily Java-oriented but with an intriguing focus on mobile development (Android), the paper offers a variety of practical suggestions on edge case

Table 3: NIST Software IA Training Resources⁵⁰

⁴⁶ Source: Michael S. Bandor, "CDRL Specific DIDs VS Only IEEE Requirements," Acquisition Community Connection, September 10, 2009. <u>https://acc.dau.mil/CommunityBrowser.aspx?id=254017</u> (accessed: April 28, 2011).

⁴⁷ Alexandra Weber Morales, "Putting some teeth into software security," SD Times, October 1, 2010. http://tinyurl.com/sd-security (accessed: April 28, 2011).

⁴⁸ The International Information Systems Security Certification Consortium, or (ISC)², grants the CISSP as well as numerous other certifications. See <u>https://www.isc2.org/Default.aspx</u> for more information.

⁴⁹ See the GIAC Web site at <u>http://www.giac.org/</u> for more information.

⁵⁰ All links are validated as of May 21, 2011. See the "Miscellaneous References" for the main NIST site.

ents/SP	800-142-	testing, input parameter tests, and the value of code assertions. However, the
<u>101006.pdf</u>		paper's math focus may limit its appeal to the general software developer.

Other useful training resources include:

- Secure Coding Guidelines for the Java Programming Language, Version 3.0 (Oracle Corporation), <u>http://www.oracle.com/technetwork/java/seccodeguide-139067.html</u>.
- The Open Web Application Security Project (OWASP) has a guide on building secure Web applications that can be downloaded from https://www.owasp.org/index.php/OWASP_Guide_Project#tab=Downloads. From IATAC's SOAR: "the OWASP Guide has served as a key source of guidance for many architects, developers, consultants, and auditors. According to OWASP, the Guide has been downloaded more than 2 million times since its publication in 2002 and is referenced by several leading government, financial, and corporate security and coding standards" (p 246).

Lead developers in particular should be given a copy of IATAC's SOAR on Software Security Assurance from the Reference List section of this paper. Additionally, these developers should subscribe to the IA Digest put out by that same organization; it features intriguing articles that are not always software development-related but are always information security-related from a DoD view.⁵¹

The result of proper training is that software developers are educated and empowered both ethically and practically to serve as an organization's most cost-effective IA asset.

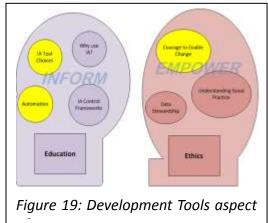
3.2.2 Development Tools and Techniques

This section shows how IA-focused development tools and techniques enhance an information system's integrity and availability while reducing rework during C&A.

3.2.2.1 Artifacts: The Fundamental Question

When confronted with a CDRL, the PM must decide how best to create, maintain, deliver, and verify the artifact demanded by the DoD customer. All too often the solution is to task a developer with the manual creation of a supporting Word document, Excel spreadsheet, PDF file, or Access database.

This approach fails because it is an out-of-band activity rather than an organic outgrowth of the software development process; thus, it invites synchronization problems and human error.



Additionally, other productive work grinds to a halt for the sake of manually producing the artifact. As technologist Alexander Egyed points out, the consequences of such an approach include "high cost, incompleteness, and even incorrectness" (EGYED, p 3).

The PM should instead view specific artifact deliverables as presentation elements rather than data elements. Though the DoD customer may require a database schema to be presented using a Microsoft Word document

⁵¹ See <u>http://iac.dtic.mil/iatac/IA_digest.html</u> for registration information and links to all of IATAC's publications.



template, it is a mistake to treat the Word document as the artifact source. More prudent is to engineer a process that generates such a Word document directly from metadata within the backing database. The remainder of this section presents redacted DoD contract deliverables from the author's own experience and discusses how they could be better created and managed using the empowered education and support advocated by IA Fusion.

3.2.2.2 Technology Stack

The specific DoD Program of Record (POR) under examination uses the Software AG webMethods Product Suite and application stack.⁵² This in no way endorses the particular use of webMethods; in fact, the author uses the Microsoft Visual Studio environment⁵³ when illustrating specific technology examples. Other application stacks such as Oracle's WebLogic application server,⁵⁴ or SAP's NetWeaver technology platform,⁵⁵ or even an opensource alternative such as RedHat's JBoss Enterprise Middleware⁵⁶ offer equally attractive software development platforms.

3.2.2.3 Example: Generating Database Schemas

2.2.2.3.1 Use Case

Consider the following example page from a required CDRL:

⁵² See <u>http://www.softwareag.com/Corporate/products/wm/default.asp</u> for more information on the Software AG webMethods Product Suite.

⁵³ See <u>http://www.microsoft.com/visualstudio/en-us</u> for more information on Microsoft's Visual Studio tool along with the ASP.NET application stack (and the Silverlight rich client interface).

⁵⁴ See <u>http://www.oracle.com/us/products/middleware/application-server/index.html</u> for more information on the Oracle application server.

⁵⁵ See <u>http://www.sap.com/platform/netweaver/index.epx</u> for more information on SAP's NetWeaver "adaptive technology for the networked enterprise."

⁵⁶ See <u>http://www.jboss.org/</u> for more information JBoss, which includes both a community edition and an enterprise edition (the enterprise edition is opensource but has a support option from the vendor).



XXXXX_DSD_AFINAL.docs

FINAL Rev A 12 April 2011

2.3 Data Service Physical Schema Definition

XXXXX.dat file consists of the following data elements:

The following table(s) define the data exchange file definition and schema definition for the service. For detailed field formats and valid values, please see the ICD from the References section above.

		Table 2-2: 1	Data Defini	tion and 8	Schema				
			Data 1	Definition					
File Format:	Fixed I	.ength							
File Format Delimiter:	NA								
Data Standard	UDF								
UDF Schema:	NA								
			XXXXXX	dat Sche	ma				
Element Name	Logical Name	Description	Element Type	Length	Order/ Position	Format	Required Element	Valid Values	РП
DODAAC	Department of Defense Activity Address Code	Department of Defense Activity Address Code (DDDAAC) is an assigned by each Service and DLA which codiens to permit shapping and for making of thiling for materiel and for making of thiling for materiel and for making of this of the second second assesses the second assesses the second second second processes and of these are DODAACs.	char	6	1:6	NA	NA	NA	м

Figure 20: Example CDRL for database schema⁵⁷

This artifact must be created manually as part of the standard software life-cycle for the POR. This requirement reflects a perception that engineering ends with management processes rather than extending to artifact creation. No manual processes should be associated with the documentation artifact; rather, the data schema should be extracted from the backend database with any required metadata (such as the file format or the file name) stored in a searchable central repository.

2.2.2.3.2 Technology Solution

The IA tenets affected here are primarily integrity and authenticity: the separation of the source data from the presentation layer make the delivered artifact suspect from the moment of its creation. A reasonable (and easy) solution would be to define a word-processing macro that can read the source data and populate the document automatically. However, managers must budget sufficient time into the project plan to analyze and engineer this automation effort.

3.2.2.4 Example: Generating Software Test Documentation

2.2.2.4.1 Use Case

The following artifact demonstrates a manual test script stored using a Microsoft Excel file:

⁵⁷ Source: Redacted DoD customer artifact.



3	Scheduled Succes	sful	10
4 5.1	IgcAVBrinit model shows process Completed	x	
5 5.2	Process Name is IgcAV8rinit	x	
6 5.3	Provider is AV in the process diagram.	×	
7 5.4	Process model is triggered by scheduled service	x	
8 5.5	Publish Async JM5 successfull; Successfully publish X JMS message to for NLAC	×	
9 5.6	Verify the files placed on the Queue matches the ThresholdNumber in config file	x	
5.7	Verify the files processed matches the MaxFilesToProcess in the config file	x	
1 5.8	Validate output file (filename and size)	x	Check in the archive (by date and hour) and sftp folde
2	Async Scheduled Suc	cessful	-W
3 6.1	IgcAsyncNlac model shows Completed	x	
6.2	Process Name is IgcAsyncNlac	x	
5 6.3	Process model is triggered by scheduled service	x	
6.4	Route SFTP successful; Successfully sftp X of X documents to NLAC	x	
7 6.5	Verify the files processed matches the MaxFilesToProcess in the confilg file.	x	Test for throttle
6.6	Verify Outbound files	x	
9	Scheduled Failed (Filenames	not ma	tched)
0 7.1	IgcAVBrinit model will not run	X	0. P. 10. 10. 10. 10. 10. 10. 10. 10. 10. 10
1 7.2	Verify that the files not matching input filter are not processed	x	
2 7.3	IgcAsyncNiac model shows process completed	x	
	Async Scheduled Failed (Directo	ory does	not exist)
8.1	IgcAVBrInit model shows process Completed	X	
8.2	IgcAsyncSmsSftp model shows process Failed	х	
5 8.3	Route SFTP error; Some documents failed to route to Niac	x	
7 8.4	Verify errorLog file (filename and size)	X	Check in the errorDump
8.5	Verify all error message variables are populated	x	
9	Async Scheduled Failed (No publish	ed files	in JMS topic)
9.1	IgcAVBrinit model will not run	x	
9.2	IgcAsyncNlac model will not run	x	
2 9.3	IgcAsyncNlac model shows process completed	x	
3	Async Scheduled Failed (invalid k	ey)
4 10.1	IgcAVBrinit model shows process Completed	×	
5 10.2	IgcAsyncNlac model shows process Failed		
6 10.3	Publish JMS successful; Successfully published X of X documents to	×	
7 10.4	IgcAsyncNlac model process shows failed	x	

Figure 21: Example Unit Test CDRL⁵⁸

The engineering that went into this deliverable can be clearly discerned: the individual steps map back to specific statements from the Capabilities Performance Statement (CPS) driving this particular effort, and the applicability as well as pass/fail results are stored in individual columns. The manual nature of the testing artifact hobbles this approach; what if the tester errs while entering data? The results of such a testing approach are, at best, suspect.

Automated software testing toolkits abound in this day and age, so the real problem is that insufficient engineering time was dedicated to test artifact generation. Aversion to taking the extra time and resources to build a test environment is understandable but short-sighted; in the long run, creating an automated testing framework may impact one specific milestone deliverable but yields a strategic asset that saves time and money overall.

2.2.2.4.2 Technology Solution

An automated software testing framework should:

⁵⁸ Source: Redacted DoD customer artifact.



- Integrate with the development environment. New tests should be automatically created for new modules. For example, development teams using the Microsoft environment can utilize the automated test capabilities of the Team Foundation Server product.
- Provide test feedback via an automated build-deploy-test module. As stable new code modules are stored to the source control system, the full software build should

Load Test Web Performan	Unit Test	Description Use a basic unit test to exercise C++, C#, or Visual Basic source code. Choosing Unit Test also lets you create ASP.NET unit tests and data-driven unit tests. A unit test calls the methods of a class, passes suitable parameters, and verifies that the returned value is what you expect. You can code unit tests by hand.
Test Name:	UnitTestI.cs	
Add to Test Project:	Create a new Visual C# test pro	yeet
		ок Cancel

occur automatically. Upon a successful build, the software should automatically invoke regression tests (first performing a simulated deployment as necessary).

- Associate tests with tasks and trouble tickets. This transparency shows conformance to contractual requirements.
- *Export output as eXtensible Markup Language (XML).* Formatting output in a common language allows for easier population of specific documentation artifact templates that the contract might require.
- Support a testing language for complex tests. Some test frameworks such as the opensource nUnit⁵⁹ or jUnit⁶⁰ packages define the testing framework as library extensions to the native language; tests are coded using the same language as the application. Other test frameworks allow tests to be coded using proprietary or standards-based languages like Business Process Execution Language (BPEL). The common requirement is that the test framework must support arbitrarily complex tests based on business rules.
- Impact program execution minimally. Consider the Software AG webMethods Developer application: this development environment does not come with a built-in test framework. Instead, developers must test programs manually in an interpreted environment and take a corresponding hit to productivity. As NIST points out, components which have worked well for months may suddenly begin to exhibit signs of

⁵⁹ See <u>http://www.nunit.org/</u> for information on nUnit.

⁶⁰ See <u>http://www.junit.org/</u> for information on jUnit.



instability when subjected to higher processing loads (NIST 800-142, p 10). If the testing framework is not sufficiently lightweight, then such load testing may be impossible to plan, implement, and verify.

Support the entire set of end-user configurations. Consider a Web application that can be run on different Web browsers. What might happen if the testing framework does not support some of those browsers? Errors that manifest themselves within one Web browser may not appear within another. For such interactive applications, the testing framework must support the same set of display mechanisms as are available to end-users. This can be especially problematic for applications that offer multiple display modes (for example, displaying both to traditional desktop environments as well as mobile environments like smartphones).

3.2.2.5 Example: Ensuring the confidentiality of Memory Resident Data

2.2.2.5.1 Use Case

Particularly rife with potential IA pitfalls is any application that must work with highly-confidential data. Such data is normally protected by defense-in-depth techniques. Encrypted databases run within servers whose operating systems are hardened to standards specified by Defense Information Systems Agency (DISA) Security Technical Implementation Guides (STIGs), and each server is further protected by at least one network firewall as well as by an Intrusion Detection System (IDS). These defenses, though valuable, are applied after-the-fact. An IA-aware software developer will take the initiative to understand the type of application data being processed and will determine cost-effective security-focused implementation measures that can be applied while the application itself is processing the data. An example of such an approach is to use "secure memory," especially when processing information read from the user such as a password or an access code.

2.2.2.5.2 Technology Solution

Memory is secured via languagespecific (and vendor-specific) techniques. The C# programming language⁶¹ as implemented by Microsoft provides the System.SecureString class, which "[r]epresents text that should be kept confidential. The text is encrypted for privacy when being used, and deleted

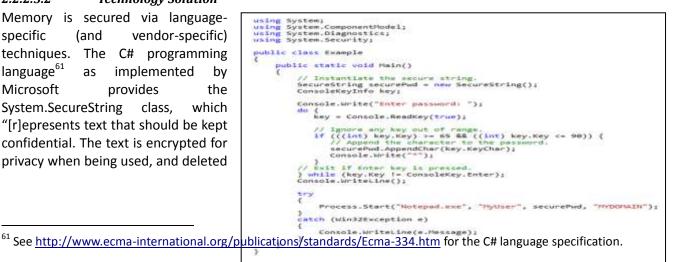


Figure 23: Secure Memory and C# (Source: MSDN)



from computer memory when no longer needed."⁶² An IA-aware software developer applies these and other inexpensive development techniques to keep the information system secure even as it is still being built.

3.2.2.6 Ethics and IA Fusion

In a structured environment, individual team members may feel that making a complaint or insisting upon following a specific security guideline risks evoking reprisal, not to mention the amount of "extra" work – in the form of meetings, write-ups, and general pushback from decision-makers – which speaking up often entails. To make it clear that ethics matter, the PM must listen to the software developers, work with them to identify true problems, and push identified problems up to the appropriate authority (for example, the designated Change Control Board or Software Advisory Board).

Unless the PM consistently stands behind the technology analyses which have been validated by the development team, the PM cannot expect any developer to challenge corporate inertia for the good of the customer. With time, and as the team is made more and more cognizant of ethics and IA, individual contributors will begin to "police themselves." Less and less direct supervision will be required of each team member's development actions to ensure conformance to quality requirements and consistent application of IA techniques.

3.2.3 Measurement

Improved software security results in a more reliable system that demands less rework; thus demonstrating a form of cost avoidance. More akin to insurance than to ROI, cost-avoidance is, unfortunately, much less of a motivator to business decision makers than ROI (STE03, p 206). To convince decision makers of the value of IA Fusion's cost-avoidant approach, one must measure results against a known baseline and justify all proposed expenses in terms of demonstrable future savings.

Software presents special problems for applying measurements; technologist Steve Bellovin argued in his 2006 address to the Metricon software security metrics convention that "[w]e can layer defenses, but once a layer is broken the next layer is exposed; it, of course, has the same problem...The strength of each layer approximates zero; adding these together doesn't help" (SOAR, 107). Despite this apparent pessimism, the SOAR goes on to argue convincingly that both qualitative and quantitative measurements are an excellent indicator of a system's security posture.

This paper reviews the state-of-the-art efforts by Carnegie Mellon University's Software Engineering Institute (SEI); the SOAR frequently references programs arising from the SEI, which in turn works closely with the DoD. The SEI's 2011 presentation "Security Measurement and Analysis" introduces a new framework and methodology for measuring software security (SEI11).

3.2.3.1 Frameworks and Protocols

The SEI's presentation lays foundational elements for security assurance measurement and analysis. A *framework* is a conceptual structure focused on relationships within a collection of components, while a *protocol* defines the sequence of activities to be performed within a method (akin to a *procedure* within ITIL).⁶³

Integrated Measurement and Analysis Framework (IMAF) and Mission-Objective-Driver (MOD) Protocol. SEI's

⁶² Microsoft Corporation, "SecureString Class," MSDN, <u>http://msdn.microsoft.com/en-us/library/system.security.securestring.aspx</u> (accessed: May 2, 2011).

⁶³ SEI11, p 18.



IMAF framework consolidates subjective and objective data into a formal analysis of a system's performance (including scaling capability). SEI proposes the following as a MOD assessment that incorporates qualitative data:

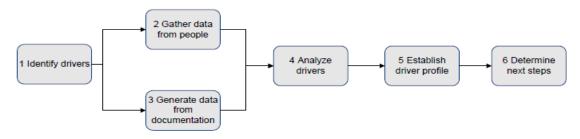


Figure 24: IMAF MOD Assessment including qualitative data⁶⁴

According to the SEI, qualitative assessments prove most effective at measuring operational security (SEI10, p 45). However, the IMAF can also be used to perform quantitative assessments. The same publication presents a use case involving Bayesian Belief Networks (BBNs)⁶⁵ to "quantitatively confirm the likelihood of occurrence of specific security entities' states as well as confirm the relationships of leading indicators among the security entities" (SEI10, p 46). One such BBN is illustrated below:

⁶⁴ Source: SEI11, p 44.

⁶⁵ Bayesian Belief Networks (BBNs) are "graphical networks that represent probabilistic relationships among variables (events or propositions). The nodes represent uncertain variables and the arcs represent the cause/relevance relationships among the variables" (STE04, p 334). BBNs are an effective way to analyze questions like "If one wakes up in the morning with a stiff neck, what is the chance that this is an early indicator of meningitis?" In other words, to discover the correlation between a stiff neck and meningitis; the answer happens to be "about 0.02%" (Charles River Analytics, <u>https://www.cra.com/pdf/BNetBuilderBackground.pdf</u>).



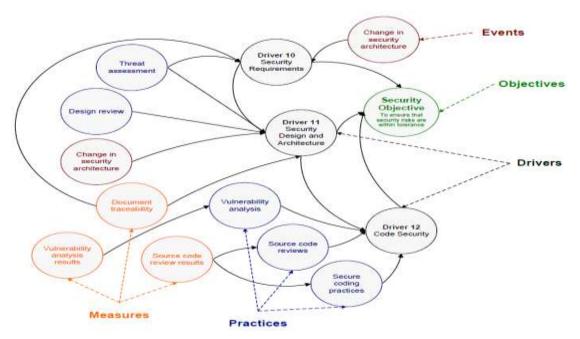


Figure 25: Bayesian Belief Network showing quantitative analysis⁶⁶

The goal of using the MOD within the IMAF is to create a quantitative measurement for the system under analysis, as shown below:

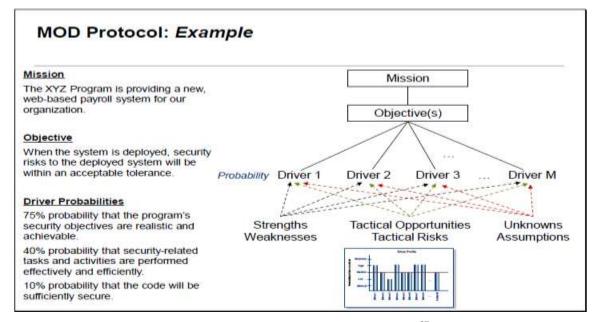


Figure 26: Example of a MOD Protocol⁶⁷

⁶⁶ Source: SEI10, p 46.

⁶⁷ Source: SEI10, p 36 (adapted).



Practice-and-Standard Mappings. Meaningful measurements map to clearly defined security objectives, which can then be abstracted into logical "drivers" (factors that have a strong influence on whether the objectives are achieved). Drivers can exist in either a success or a failure state (zero / one); the goal is to determine the relative likelihood for a driver to be in a given state based on a varying set of controls. Drivers are always dependent upon interactions among a set of measurements.

A single driver can relate to multiple objectives as shown in the preceding figure. BBNs can then be constructed based on interviews with stakeholders and Subject Matter Experts (SMEs) to determine the circumstances, conditions, or events that will put a driver into its successful state. The goal is to simplify potential outcomes into statements such as, "Driver X has a Y percent chance of being in a successful state given the set of controlling factors *a*, *b*, and *c*." Decision makers can then prioritize funding based on success probabilities displayed by the resulting BBN.

3.2.3.2 Methodologies

Security practitioners can apply the following practical methodologies:

The Software Security Review (SSR). SEI defines this as "a method conducted by independent teams to assess the security characteristics of software-reliant systems" (SEI11, p 29). This corresponds closely with the software code reviews and peer reviews previously discussed within this paper. SEI extends this concept to include the entire supply-chain life-cycle, making SSR of especial use to acquisition managers. The SSR's phases and activities are shown below:

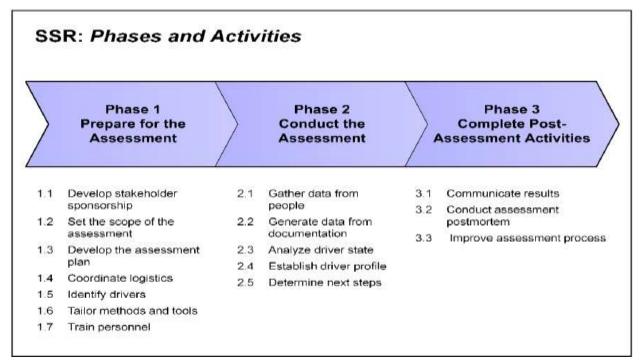


Figure 27: SRR Phases and Activities⁶⁸

⁶⁸ Source: SEI11, p 31.

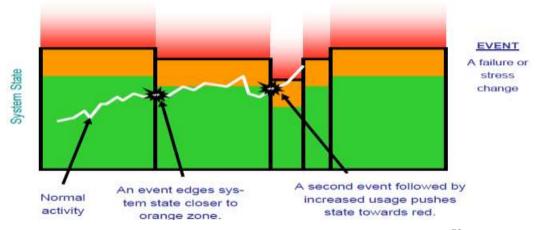


First, the practitioner prepares for the assessment in collaboration with the *project sponsor* (in an Agile project this would be the Product Owner). Such support must be provided by a particular individual with the authority to fund and drive the assessment. The scope of the assessment must be set upfront so that personnel can be trained to conduct it properly. (This facilitated approach brings to mind Thomas Peltier's Facilitated Risk Analysis and Assessment Process, or *FRAAP*, which the author has discussed in a prior paper.)⁶⁹

IMAF and MOD can be employed to analyze the data once it has been gathered. Each identified driver's completed analysis must then be communicated to the team. SEI advocates assigning the likelihood of the driver's reaching a successful state to one of the following qualitative labels: Yes (definite), Likely Yes, Equally Likely, Likely No, and No (definite). The set of analyzed drivers can then be displayed within a dashboard as a Key Performance Indicator (KPI).

Multi-view Decision Making (MVDM). This methodology enables insight into both single systems as well as systems-of-systems. MVDM focuses on the early phases of the life-cycle (acquisition, development, and deployment), and thus has particular relevance to IA Fusion's emphasis on the software build phase. While SSR views the breadth of an organization's security mission and posture, MVDM plumbs the depth of individual programs and even individual projects.

SEI created an MVDM workshop describing a notional pilot to improve Close Air Support within the U.S. Army. In this workshop, the ability of "stress events" to escalate into failure modes was clearly illustrated. Complex systems fail because the interaction of different components makes it difficult to create test environments without performing the MVDM analysis.



*Figure 28: Multiple event failures collude to cause failures*⁷⁰

3.2.3.3 Summary

Software security measurement and analysis can establish, specify, and measure confidence in a software system's ability to meet operational needs while retaining a required security level. The SEI's innovative approach builds on a foundational set of measurement concepts. This paper's "Appendix A: Recommendations

⁶⁹ See <u>https://blogs.rividium.com/file.axd?file=2011%2F2%2FRisk+Management+and+Methodologies.pdf</u> for the author's comparison of three risk assessment methodologies, including the FRAAP.

⁷⁰ Source: SEI09, p 15.



by Cost and Priority" suggests forward steps and rough estimates that can be used to implement this methodology within a DoD contractor's software development effort.

4.0 Concluding Remarks

This paper has analyzed how a DoD contractor's software developers (individual contributors) can be educated, empowered, and enabled to integrate IA into the software development process via an approach designated "IA Fusion." Reactive attempts to layer IA on top of existing processes are more expensive and less effective than fusing IA into those processes. Even DoD's own publications perpetuate the faltering approach of the former rather than the smooth integration of the latter by consistently placing IA under the purview of IA officers and C&A assessors. Under IA Fusion, software developers are trained to orient their myriad day-to-day decisions towards IA principles and best practices.

IA Fusion is firmly grounded in DoD policy; specifically, the systems engineering and C&A processes that all DoD contractors must follow as exemplified by DoD 8570.01-M ("IA Workforce Improvement Program"), ISO/IEC 12207 ("Information Technology: Software Life-cycle Processes"), DoD 8500.2 ("IA Implementation"), and DoD 8510.01 ("DoD Information Assurance Certification and Accreditation Process"). A DoD contractor need not be concerned that taking the IA Fusion approach will place its contract deliverables at risk.

IA Fusion reduces costs by encouraging individual contributors to remain cognizant of IA throughout a project's life-cycle. Specific recommendations made by this paper include:

- *Championship.* Build esprit-de-corps within the software development team.
- *IA Control Awareness.* Educate software developers on practical ways to base their everyday decisions on IA principles and practices.
- Secure Coding. Craft defensive and reliable code solutions.
- Automated Artifact Generation. Generate accurate contract deliverable artifacts with minimal lost productivity.
- *IMAF Implementation*. Engage with SEI to implement leading-edge security measurement solutions.

This paper presents "Appendix A: Recommendations by Cost and Priority" with cost and schedule estimates for the preceding high-level recommendations.

IA Fusion adds real value throughout the organization. By establishing measurement baselines against quantifiable elements such as the time necessary for an interim release (or "Sprint" if using Agile project management), the PM can show that leveraging the talents of the software development team results in quantifiable savings. Additionally, the organization is building what ITIL calls a *Capability*: "an intangible Asset of an Organization."⁷¹ Capabilities are the most valuable assets within an organization's intellectual property; furthermore, a Capability cannot be bought, but must be developed over time. Capabilities differentiate the holder from its competition and sharpen the organization's competitive edge (both strategic and tactical).

In today's cost-conscious DoD environment, using IA Fusion to leverage the untapped potential of software developers helps to deliver the best possible value to one's government customer, and from there to the

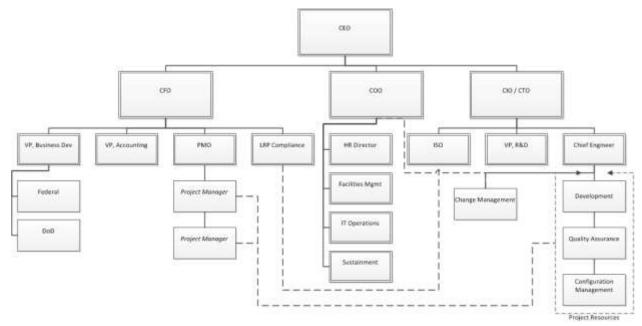
⁷¹ Source: ITIL Definitions, <u>http://www.knowledgetransfer.net/dictionary/ITIL/en/Capability.htm</u> (accessed: May 19, 2011).



ultimate end-customer: the U.S. Soldier.

Appendix A: Recommendations by Cost and Priority

This paper has analyzed a number of ways by which software developers can be integrated ("fused") into the IA process. This Appendix sorts estimates by cost and by priority for the PM's consideration. The notional DoD contractor used as a basis in the chart below is a strong-matrixed⁷² organization whose development group operates under the overall direction of the Chief Engineer with significant input and control from individual project managers, along with a dedicated resource commitment to each project effort.



⁷² The PMI defines a *strong matrix* as having "many of the characteristics of the projectized organization, and can have full-time project managers with considerable authority and full-time project administrative staff" (PMI, p 29). A *projectized* organization has staff reporting exclusively to the PM; in the author's drawing, the development staff works under the Chief Engineer but the PM has considerable authority to optimize project processes. A strong PM role is essential to the concept of IA Fusion, which requires project contributors at the edge to be engaged and proactive in improving specific projects in order to obtain organization-wide benefits.



Figure 29: Strong-Matrixed Organizational Outline⁷³

A.1: Summary

The following table shows the estimated costs for each recommendation made by the paper. Estimates should be considered as rough orders of magnitude, with -50% to +150% being the possible range of actual implementation costs.

Recommendation	Costs	Total Days
Automated Artifact Generation	\$46,588	97
IMAF Implementation	\$43,059	69.5
Project Championship	\$33,647	17
Secure Coding	\$32,111	30.5
IA Control Awareness	\$26,563	45
Totals:	\$181,968	259

A.2: Recommendations by Cost (Highest to Lowest)

A.2.1: Automated Artifact Generation

To generate automated artifacts requires both processes and tools. This section examines four aspects of artifact generation:

- *Project Management Traceability.* The ability to track work performed back to development tasks and contract requirements.
- *Design Documentation.* The supporting documents that provide a visual guide to the structure and architecture of an implemented system.

⁷³ Drawing by the author.



- *Test Results Documentation.* The verifiable proof that a defined set of tests were run.
- Source Code Documentation. The library of software functions and modules that the developed system contains, and the relationships between those functions and modules.

A.2.1.1: Scope

Automated artifact generation tools exist as Commercial-off-the-Shelf (COTS) products for numerous platforms and range in scope from opensource to highly proprietary. This section presents selected tools for the Microsoft family of programming products; specifically, the Microsoft Visual Studio development environment (VS2010) and the Microsoft Team Foundation Server (TFS) source control and project integration environment.

A.2.1.2: Project Management Traceability

Project management traceability helps in accounting for the true cost of a Work Breakdown Structure (WBS) item by allowing for the tracking of specific activities associated with that WBS item. The IBM DOORS⁷⁴ product works with the VS2010 and the TFS environments, and is used as a notional baseline for cost and implementation estimates. The DOORS product is an enterprise-class solution that:

- Encourages stakeholder engagement and collaboration amid its comprehensive requirements management environment
- Offers a Web browser interface.
- Manages requirements changes via customizable change control workflows, and enables informal requirements discussions with DOORS Discussions (built-in messaging functionality).
- Integrates not just with Microsoft products but with other industry-standard solutions (particularly IBM's Rational series of products).

⁷⁴ See the IBM Web site at <u>http://www.ibm.com/software/awdtools/doors/</u> for more information on the DOORS product (accessed: May 21, 2011).



"IA FUSION" FOR A DOD CONTRACTOR

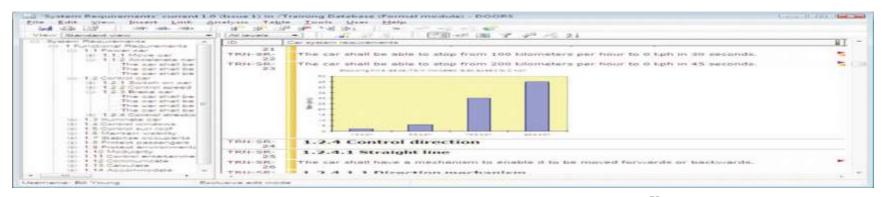


Figure 30: Example IBM DOORS Project Management Integration⁷⁵

A.2.1.3: Design Documentation

This paper has examined DoD documentation requirements; the assumption is that the end-result is a system that can be delivered to the DoD customer to run on a secure military network. The DIACAP allows systems to be certified and accredited to run on such networks, and an innovative company (I-Assure) offers a free DIACAP artifact generation program available from its home page.⁷⁶ This toolset can create a System Identification Profile, DIACAP ScoreCard, and a Plan of Actions and Milestones (POAM).

A.2.1.4: Test Results Documentation

The Software Test Plan (STP) described as a standard contract artifact in this paper is an excellent candidate for automation. One such COTS product that provides an enterprise-class solution is the TestComplete system, which integrates tightly with VS2010 and can generate customized test outputs. As system testers and developers define unit tests, these tests can be automated and the results stored for comparison against a test baseline. Both Agile and Waterfall projects can benefit from this capability.

⁷⁵ Source: "IBM Rational Doors" data sheet, <u>http://public.dhe.ibm.com/common/ssi/ecm/en/rad14037usen/RAD14037USEN.PDF</u> (accessed: May 10, 2011).

⁷⁶ See the I-Assure pages at <u>http://www.i-assure.com/products.htm</u> for a link to the DIACAP Toolset product (accessed: May 12, 2011). Requires free registration to download.



"IA FUSION" FOR A DOD CONTRACTOR

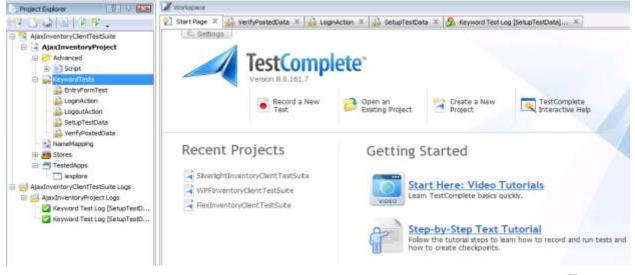


Figure 31: Example Integrated Software Testing Tool with Customizable Output⁷⁷

A.2.4.5: Source Code Documentation

The VS2010 development environment discussed in this paper allows software commenting out-of-the-box. Developers can use a set of "documentation tags" loosely based upon JavaDoc documentation standards.⁷⁸ These "tags" allow the developer to indicate the purpose of specific software modules and functions, input and output parameters, and expected usage notes and assumptions. Additionally, some testing tools allow "tag extensions" to define test criteria for individual program functions (thus aiding in unit testing). From a contract deliverable view, the real value of source code documentation is in providing a turn-key methodology for generating required contract artifacts without requiring a separate manual process.

⁷⁷ Source: "TestComplete 8" product page, <u>http://www.automatedqa.com/products/testcomplete/</u> (accessed: May 10, 2011).

⁷⁸ See the Oracle JavaDoc tool page for more information, <u>http://www.oracle.com/technetwork/java/javase/documentation/index-jsp-135444.html</u> (accessed: May 10, 2011).



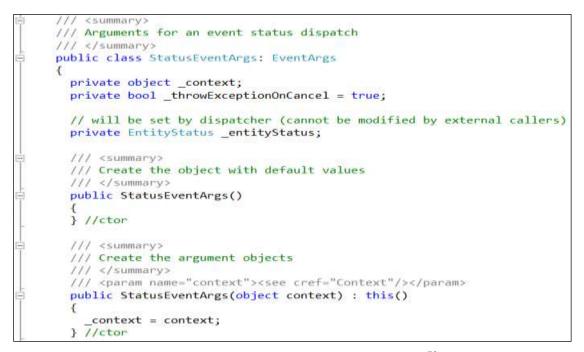


Figure 32: Example C# Documentation Tags⁷⁹

Numerous COTS tools exist to supplement the out-of-the-box functionality provided by VS2010. This paper uses the highly-regarded VSDocMan tool⁸⁰ as a typical enterprise-class product for creating project artifacts in support of IA Fusion because of its:

- Flexible output formats (especially useful to repackage output to meet contract artifact formats);
- Automatic commenting (useful to enforce a code commenting policy on individual developers); and,
- Comment editors and class diagrams (allows more elaborate comment support, such as tables of figures).

⁷⁹ Created by the author from a live project.

⁸⁰ See the VSDocMan home page at <u>http://www.helixoft.com/vsdocman/overview.html</u> for more information (accessed: May 21, 2011).



A.2.4.6: Example Cost Matrix

Automated Artifact				Costs		Total
Туре	Activity	Detail	Roles	Calculation	Value	Days
	Management Integration	Setup project plans for DOORS / TFS integration	PM	5 x 261	\$1,305	5
		Purchase DOORS (floating license) ⁸²		[purchase price]	\$9,900	
	Installation	Setup	Config Mgmt Specialist	5 x 246	\$1,230	5
		Integration with project management tools	Business Analyst (Database)	10 x 211	\$2,110	10
Project Management Traceability	Training		Team	1 day x 10 x 196	\$1,960	1
indecubility	Totals:				\$16,505	16
	Ongoing maintenance (annual)	System Administration	Config Mgmt Specialist	15 x 246	\$3,690	15
		Software license renewal		[purchase from vendor]	\$1,955	
	Feedback	Ongoing (included in the Project Championship closed loop)	Team			
	Maintenance Totals:				\$5,645	15
		Create IA plan	PM	4 x 261	\$1,044	4
Design Documentation	IA Analysis		IA Analyst	8 x 261	\$2,088	8
		Software (DIACAP Toolset)	[n/a]	[free]		
	Control Mapping	Perform mapping	IA Analyst	8 x 261	\$2,088	8
	8	Assist IA Analyst with mapping	Development SMEs	4 days x 4 SMEs x 403	\$6,448	16

Table 5: Estimated Costs for Artifact Generation⁸¹

⁸¹ Costs are expressed as units (days) multiplied by estimated salary. See Appendix "A.4: Selected Employee Costs" for salary estimates.

⁸² Source: IBM Web site <u>http://www-142.ibm.com/software/dre/ecatalog/detail.wss?locale=en_US&synkey=F036918J78119U24</u> (accessed: May 12, 2011).



"IA FUSION" FOR A DOD CONTRACTOR

Automated Artifact				Costs		Total
Туре	Activity	Detail	Roles	Calculation	Value	Days
	POAM Creation	Create plan based on gap analysis (mapping)	PM	3 x 261	\$783	3
		Verify plan and enter to DoD databases as necessary	IA Analyst	3 x 261	\$783	3
	Totals:				\$13,234	42
		Design test cases from requirements	Team	Approx 10% of project time		
	Test Planning	Purchase TestComplete (floating license) ⁸³			\$4,499	
Test Results		Install / Configure Software	Config Mgmt Spec	3 x 246	\$738	3
Documentation	Baseline creation	Analysis and definition	QA Engineer	5 x 244	\$1,220	5
	Report Creation	Customization to DoD contract requirements	QA Engineer	10 x 244	\$2,440	10
	Totals:				\$8,897	18
		Documentation Standards	PM			
	Discription	Documentation standards	SMEs	2 days x 4 SMEs x 403	\$3,224	8
Source Code	Planning	Software purchase (ten licenses, fixed price) ⁸⁴		10 license x 155 per license	\$1,550	
Documentation		Software configuration	Config Mgmt Spec	3 x 246	\$738	3
	Report Creation	Customization to DoD contract requirements	QA Engineer	10 x 244	\$2,440	10
	Totals:				\$7,952	21
Grand Totals (<i>no</i> <i>maintenance</i>):					\$46,588	97

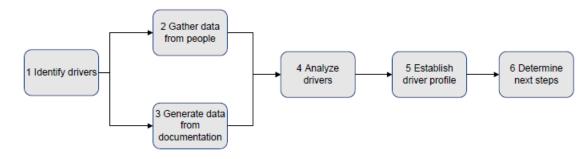
A.2.2: IMAF Implementation

This paper analyzed the Integrated Measurement and Analysis Framework (IMAF) from the Software Engineering Institute (SEI) of Carnegie Mellon University. This framework provides measures for secure software systems and the SEI team is looking for organizations to work with

⁸³ Source: SmartBear Software pricing page, <u>http://www.automatedqa.com/products/testcomplete/ordertestcomplete/</u> (accessed: May 10, 2011).

⁸⁴ Source: HelixSoft pricing page, <u>http://www.helixoft.com/common/buy-helixoft-products.html</u> (accessed: May 10, 2011).





them in demonstrating the framework's effectiveness, especially the SEI's "standard set of drivers for security software" (SEI10, p 36).

Figure 33: The IMAF Qualitative Assessment⁸⁵

The following table breaks down each step from the above. The first step (Identify Drivers) is taken as complete, where the goal is to review the SEI's set of 17 standard security software drivers.

			Costs		Total
Activity	Detail	Roles	Calculation	Value	Days
	Mission Statement	Executives	[assumed done]	0	0
Identify Drivers	Objectives	Executives	[assumed done]	0	0
	Create Drivers (use SEI's standard set of 17 drivers)	SEI	[n/a]	0	0
Gather Data from People	Identify Stakeholders	VPs and Chief Engineer	1 day x 6 persons x 423	\$2538	1

Table 6: IMAF Implementation Estimates⁸⁶

⁸⁵ Source: SEI10, p 44.

⁸⁶ Costs are expressed as units (days) multiplied by estimated salary. See Appendix "A.4: Selected Employee Costs" for salary estimates.



"IA FUSION" FOR A DOD CONTRACTOR

			Costs		Total
Activity	Detail	Roles	Calculation	Value	Days
		PM	4 days x 338	\$1,352	4
		PM	6 x 211	\$1,266	6
	Perform Interviews	Stakeholders	.25 days x 10 (estimate) x 403 (director level)	\$1,007	2.5
	Organize Results	PM	2 x 338	\$676	2
		PM	3 x 338	\$1,014	3
	Locate Documentation	SMEs	1 day x 10 (estimate) x 403 (director level)	\$4,030	1
	Identify Tools (including email-based survey tool)	PM	3 x 338	\$1,014	5
Generate Data from		SMEs	5 days x 3 (estimate) x 403	\$6,045	3
Documentation	Run Tools and Gather Data	PM	5 x 338	\$1,690	5
		Config Mgmt Spec	5 x 246	\$1,230	5
		SMEs	3 days x 3 (estimate) x 403	\$3,627	3
	Organize Results	PM	2 x 338	\$676	2
		Technical Editor	5 x 192	\$960	5
	Build qualitative response checklist (SEI recommends six possible responses from "Yes" to "Don't Know")	PM	1 x 338	\$338	1
		SMEs	1 day x 3 (estimate) x 403	\$1,209	1
Analyze Drivers	Determine value driver criteria (mapping responses both to success	PM	1 x 338	\$338	1
	probability and failure probability; SEI recommends the qualitative values "Minimum," "Low," "Medium," "High," and "Maximum").	SMEs	1 day x 3 (estimate) x 403	\$1,209	1
	Organize results	PM	1 x 338	\$338	1
		PM	1 x 338	\$338	1
	Map drivers to organizational goals and objectives	SMEs	1 x 3 (estimate) x 403	\$1,209	1
Establish Driver Profile		PM	2 x 338	\$676	2
	Create survey questions	SMEs	2 x 3 (estimate) x 403	\$2,418	2



"IA FUSION" FOR A DOD CONTRACTOR

			Costs		Total
Activity	Detail	Roles	Calculation	Value	Days
	Perform surveys	Stakeholders	.25 days x 10 (estimate) x 403	\$1,007	.25
	Calculate responses to selected scale (values below mid-line are	PM	1 x 338	\$338	1
	considered "likely in their failure state").	SMEs	1 x 3 (estimate) x 403	\$1,209	1
	Prioritize results	PM	1 x 338	\$338	1
		SMEs	1 x 3 (estimate) x 403	\$1,209	1
Determine Next Steps	Create POAM	PM	2 x 338	\$676	2
		PM	4 x 338	\$1,352	4
Present to N	Present to Management	VPs and Chief Engineer	.5 x 6 x 423	\$1,269	.5
		Executives	.25 x 4 x 463	\$463	.25
Totals:				\$43,059	69.5

A.2.3: Project Championship

Championship: Project managers champion their software developers within the organization.⁸⁷

The following diagram suggests one model for championing developers:

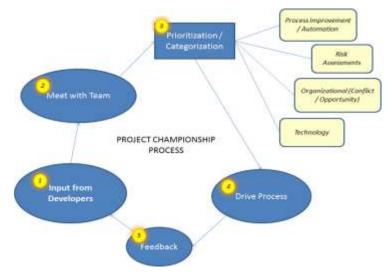


Figure 34: Notional Project Championship Process⁸⁸

- 1. Input from developers. The PM needs to develop relationships and elicit input from individual software developers.
- 2. Meet with Team. The PM must have scheduled and formal sessions where input can be discussed and clarified.
- 3. *Prioritization / Categorization.* The PM reviews the clarified Team input and determines the potential impact of the recommendations under review. Potential categories into which these recommendations could be classed include:

⁸⁷ Restatement of the "Project Champion" role advocated by this paper.

⁸⁸ Drawing by the author.



- *Process Improvement / Automation:* Automate artifact generation; eliminate redundant activities; and, construct new processes by which to generate software deliverables.
- *Risk Assessment:* Identify IA concerns in the code such as cleartext passwords; raise discovered issues found with COTS products and tools to the appropriate advisory board.
- Organizational: Identify conflicts with other departments or external entities; build upon opportunities for partnering.
- *Technology:* Provide expert judgment and / or feedback on possible improvements to the technology products being used within a project.
- 4. *Drive Process.* The PM takes the prioritized and categorized ideas and drives them upward through the organization. The goal is to show decision makers how the improvements can save time or resources. The PM should integrate with the Change Control Board (CCB), Software Engineering Review Board (SERB), Risk Management Board (RMB), and many other groups.
- 5. *Feedback.* The PM ensures that the software development Team (especially the individual contributor who made the original suggestion) receives updates on the forward motion of the suggestion. When a suggestion is either voted down or put on hold by decision makers, the PM shares that information and determines why it happened.

The result is a self-informing loop that ensures individual contributors are aware of the influence they have to control their own environment (known as "locus of control"). Such empowerment heightens the Team's morale and performance (BOS09, p 1322).

			Costs		Total
Activity	Detail	Roles	Calculation	Value	Days
Input from Developers	Team-building half-days to build rapport	Team	(.5 x 6 (bi-monthly)) x 10 x 196	\$5,880	3
Meet with Team	Formal "continuous process improvement" meetings with analysis; these are <i>not</i> status meetings.	Team	(.25 x 12 (monthly)) x 10 x 196	\$5,880	3
Prioritization / Categorization	Analysis time by PM	PM	(.25 x 12 (monthly)) x 338	\$1,014	3
Drive Process	Integrate with CCB	ССВ, РМ	(.25 x 12 (monthly)) x 6 x 338	\$6,084	3
	Integrate with SERB	SERB, PM	(.25 x 12 (monthly)) x 6 x 338	\$6,084	3

Table 7: Project Championship Costs and Timeline⁸⁹

⁸⁹ Costs are expressed as units (days) multiplied by estimated salary. See Appendix "A.4: Selected Employee Costs" for salary estimates.



			Costs		Total
Activity	Detail	Roles	Calculation	Value	Days
	Integrate with RMB	RMB, PM	(.25 x 12 (monthly)) x 6 x 338	\$6,084	3
	Present to Decision-Makers ("VP"-level)	VP	(.25 x 4 (quarterly)) x 3 x 423	\$1,269	1
		PM	(1 x 4 (quarterly)) x 338	\$1,352	4
Feedback	Ongoing; included in the Meet with Team and Input from Developers	Team			
Totals:				\$33,647	17

A.2.4: Secure Coding

Training the developers to use secure coding techniques can rapidly provide a measurable return on investment. Software runs more reliably, defects are detected more quickly, and root causes can be diagnosed without the need to muddle around with "educated guessing" on the part of a developer working under pressure to resolve a software incident. This paper's "Integrating IA Fusion" section identified many software development training resources; this section continues in that vein by providing a cost estimate for implementing a secure coding environment within a specific project. Microsoft provides C#-specific secure coding resources at its Secure Coding Guidelines page (<u>http://msdn.microsoft.com/en-us/library/d55zzx87(v=vs.90).aspx</u>, accessed May 10, 2011). Additionally, the Microsoft Secure Development Life-cycle is available at <u>http://www.microsoft.com/security/sdl/discover/default.aspx</u> (accessed May 10, 2011) and provides a model that can be used to estimate costs.



"IA FUSION" FOR A DOD CONTRACTOR



Figure 35: Microsoft Secure Development Life-cycle⁹⁰

⁹⁰ Source: Microsoft Security Development Life-cycle, <u>http://www.microsoft.com/security/sdl/discover/default.aspx</u> (accessed May 10, 2011). Highlights by the author.



For this cost estimate, only the portion of the Secure Development Life-cycle that corresponds to the "Training and Resources (Education)" section of this paper is analyzed.

			Costs		Total
Activity	Detail	Roles	Calculation	Value	Days
	Security Classes ⁹² (at third-party location)	Team	2.5 days x (400/day (training) + 196/day (salary)) x 10	\$14,900	2.5
Core Security Training	Specialized CBT modules	SMEs	4 x 2 x 403	\$3,224	8
		Team	1 x 10 x 196	\$1,960	1
	Refresher	Team	.25 days x 4 (quarterly) x 10 x 196	\$1,960	1
	Peer reviews	Business Analyst (Database)	6 x 211	\$1,266	6
Verification	Code analysis (coverage metrics)	IA Analyst	.25 days x 6 x 261	\$391	.25
		Team	.25 days x 6 x 8 x 196	\$2,352	.25
		SMEs	.25 days x 6 (bi-monthly) x 4 x 403	\$2,418	1.5
		PM	3 x 338	\$1,014	3
Release	Incident response planning	IA Analyst	3 x 338	\$1,014	3
		SMEs	1 days x 4 x 403	\$1,612	4
	Final Security Review (elements included in Project Championship)				
Totals:				\$32,111	30.5

Table 8: Secure Coding Cost Estimates⁹¹

⁹¹ Costs are expressed as units (days) multiplied by estimated salary. See Appendix "A.4: Selected Employee Costs" for salary estimates.

⁹² LifeTech Academy used for notional costing purposes, <u>http://www.lifetechacademy.com/training_cost/training_cost_page.html</u> (accessed May 10, 2011).

A.2.5: IA Control Awareness

Software developers need to be aware of IA controls so that they can incorporate IA into their day-to-day decision-making as they execute specific project tasks. For example, IA-aware developers would identify temporary data file storage as a prime candidate for an improved security solution. (One solution might be to encrypt the data files and apply extremely restrictive access control lists (ACLs) to those files.) IA control awareness training can be delivered via computer-based training (CBT) modules.

			Costs		Total
Activity	Detail	Roles	Calculation	Value	Days
	Align project deliverables to IA policies (organizational and DoD)	Information Security Off	5 x 261	\$1,305	5
		PM	3 x 338	\$1,014	3
Material Development	Technical Editor	5 x 192	\$5,880	3	
	Create customized training per DoD 8570.01-M (e.g. database / developer / administrator roles)	PM	5 x 338	\$1,690	5
		Web Developer (CBT)	10 x 196	\$1,960	10
		Technical SMEs	4 days x 4 roles x 403	\$6,448	16
		PM	2 x 338	\$676	2
Planning	Records management	Business Analyst (Database)	6 x 211	\$1,266	6
	Training schedules	PM	1 x 338	338	1
	Management Approval	Chief Engineer	.25 * 423	\$106	.25
Initial Training	CBT delivery	Team	2 days x 10 persons x 196	\$3,920	2
Ongoing Training	CBT delivery	Team	.25 days x 4 (quarterly) x 10 persons x 196	\$1,960	10
Feedback	Ongoing (included in the Project Championship closed loop)	Team			
Totals:				\$26,563	45

Table 9: IA Control Awareness⁹³

⁹³ Costs are expressed as units (days) multiplied by estimated salary. See Appendix "A.4: Selected Employee Costs" for salary estimates.



A.3: Recommendations by Priority

This section prioritizes recommendations in the order that they should be implemented by the PMO.

Item	Reasoning
Championship	To build esprit-de-corps to and ensure that IA Fusion can be successful, the software developers must trust and believe in their project management team.
IA Control Awareness	Software developers cannot be expected to implement security concepts as an organic part of the software development process without first understanding the IA control requirements. As the software code base changes, integrated builds and regression tests can detect problematic software modifications. These modifications can be tied back to the software developer in charge of a particular module. Implementation of an automated toolset can help the developer to align each "code commit" (check-in of a new software version) to contractual line items and decomposed tasks, thus improving the project's overall quality assurance.
Secure Coding	Day-to-day decisions made by individual software developers begin to take into account IA requirements. Software developers begin to code more defensively, with an eye towards validating and protecting data accepted from external sources (users and / or other systems). Software developers begin to police themselves by critically analyzing each other's work during peer reviews.
Automated Artifact Generation	Once the team trusts management and understands the requirements for IA Fusion, the organization can start to realize IA Fusion's cost-reduction benefits in the form of correctly automated IA artifact generation. The software development team works with the project manager and the IA officers to identify automation opportunities and to increase artifact quality. Automated tools detect common problems such as untested code sequences or code not written to agreed-upon organizational standards, thus ensuring the production of high-quality documentation artifacts.
IMAF Implementation	The SEI seeks partners to implement the IMAF software security measurement framework in production environments. This framework provides the set of sequences and steps necessary to ensure rigor within the software development effort and to demonstrate IA Fusion's measurable benefits to organizational decision makers.

Table 10: Recommendations by Priority



A.4: Selected Employee Costs

The following table lists average salaries for selected roles. All values are rounded to the nearest whole number and averaged between the low and high values available from the data source. In order to provide the reader with a more complete reference this paper includes common roles in addition to the specific employee roles identified in the cost estimates above.

Position	Average Annual Salary	Average Daily Cost
IT Executive (CEO / CFO / COO / CIO)	\$120,481	\$463
IT Vice President (includes PMO, Chief Engineer, etc.)	\$110,000	\$423
IT Project Manager	\$88,000	\$338
Regulatory Compliance Manager (LRP Compliance)	\$77,000	\$296
HR Director	\$84,000	\$320
Facilities Manager	\$62,000	\$238
IT Operations Manager	\$90,000	\$346
Configuration Management Specialist (Config Mgmt Spec)	\$64,000	\$246
QA Engineer	\$63,500	\$244
Information Security Officer	\$82,000	\$315
Information Assurance Analyst	\$68,000	\$261
Technical Editor	\$50,000	\$192
Senior Software Engineer (typically Subject Matter Experts for all disciplines)	\$105,000	\$403
Business Analyst (Database)	\$55,000	\$211
Average Information Technology Worker (including average Web developer)	\$51,000	\$196

Table 11: Selected	Employee	Costs for	a 2,000-perso	on Company ⁹⁴

⁹⁴ Salary estimates come from <u>http://www.payscale.com/research/US/</u> (accessed: May 14, 2011).



Appendix B: MIL-STD-498, ISO/IEC 12207, and IEEE/EIA 12207: What is the Standard?⁹⁵

Some naming confusion exists within the DoD software development standards; most noticeably, the 12207 guidance document. In 1988 the DoD released DOD-STD-2167A ("Defense System Software Development"), which in 1994 evolved into MIL-STD-498: MIL-STD-498 contained the twenty-two standard DID contract deliverables and a wealth of practical implementation advice. Both the software industry and international standards bodies recognized the value of DoD's methodology; as such, the DoD sought to de-militarize this standard with J-STD-016-1995 in September 1995 ("Software Life-cycle Processes, Software Development"). Almost simultaneously came ISO/IEC 12207 in August of 1995, with Institute of Electrical and Electronics Engineers / Electronic Industries Alliance (IEEE/EIA) Standard 12207 ("Software Life-cycle Processes") materializing shortly afterwards in March 1998 (Guides) and April 1998 (Standards).

Dr. Raghu Singh was instrumental throughout this evolution, serving as Chair for the MIL-STD-498 working group, Editor for the ISO/EIC 12207, Co-Chair for the J-STD-016-1995 working group, and Co-Chair for the IEEE/EIA 12207.0-1996 working group. The common history of these documents is illustrated below:

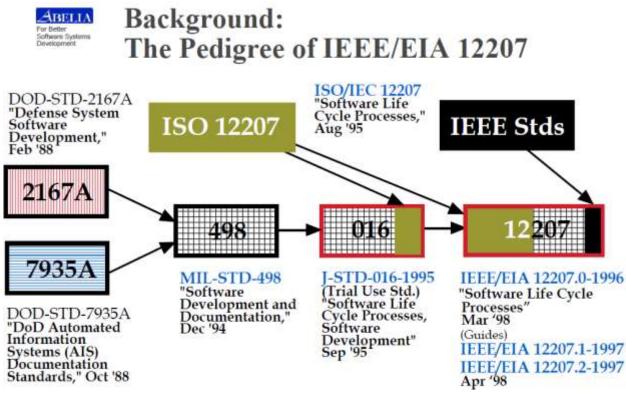


Figure 36: The Pedigree of IEEE/EIA 12207⁹⁶

⁹⁵ This section was created from Dr. Lawrence Gray's 1999 presentation "A Comparison of IEEE/EIA 12207, ISO/EIC 12207, J-STD-016, and MIL-STD-498 for Acquirers and Developers" at http://www.abelia.com/docs/122_016.pdf. It has been used with his permission.

⁹⁶ Used with permission from Dr. Gray's presentation (p 5).



Noted DoD software security expert Dr. Lawrence Gray describes the difference between the standards thus: "ISO/EIC 12207 and IEEE/EIA 12207 are about the software development life-cycle; in contrast, MIL-STD-498 and J-STD-016 are about what developers do." The IATAC notes that IEEE Standard 1074-2006 ("Developing Software Project Life-cycle Processes") "add[s] support for prioritization and integration of appropriate levels of security controls into software and systems" (SOAR, p 253).

Owing to the practicality and wide availability of the MIL-STD-498 / J-STD-016 standards, along with their attendant DIDs, these standards are unlikely to be supplanted completely in common usage by the newer IEEE / EIA standards.



Appendix C: Acronyms and Abbreviations

Where an acronym has meaning in a specific context (for example, with regard to a specific NIST publication), that context is noted parenthetically.

ACL	Access Control List (file access)
АКО	Army Knowledge Online (DoD social networking; <u>https://www.us.army.mil/</u>)
BBN	Bayesian Belief Network
BPEL	Business Process Execution Language
CAC	Common Access Card
CBT	Computer-Based Training
ССВ	Change Control Board
CDRL	Contract Data Requirements List
CIO	Chief Information Officer
CISSP	Certified Information Systems Security Professional
СМ	Configuration Management
CND	Computer Network Defense
CNSS	The Committee for National Security Systems
COTS	Commercial-off-the-Shelf (software)
CPS	Capabilities Performance Statement
DAG	Defense Acquisition Guide
DD	Department of Defense (Form)
DIACAP	DoD IA Certification and Accreditation
DID	Data Item Description (MIL-STD-498 documentation standard)
DISA	Defense Information Systems Agency
DoD	Department of Defense
EMD	Engineering and Manufacturing Development Phase
FRAAP	Facilitated Risk Analysis and Assessment Process
GIAC	Global Information Assurance Certification
GIG	Global Information Grid
IA	Information Assurance
IASE	Information Assurance Support Environment
L	



	1
ΙΑΤΑϹ	Information Assurance Technology Analysis Center
IDS	Intrusion Detection System
IMAF	Integrated Measurement and Analysis Framework (methodology created by SEI)
ISO/IEC	International Organization of Standards / International Electrotechnical Commission
ISSO	Information System Security Officer (NIST SP 800-64 reference)
ITIL	IT Infrastructure Library
КО	Contract Officer
КРІ	Key Performance Indicator
KS	Knowledge Service (DIACAP online collaboration and information portal)
MAC	Mission Assurance Category
MIL-STD	Military Standard
MOD	Mission-Objective-Driver (from the SEI IMAF methodology)
MVDM	Multi-view Decision Making (from the SEI IMAF methodology)
NDIA	National Defense Industrial Association
NIST	National Institute of Standards and Technology
OWASP	Open Web Application Security Project
PII	Personally Identifiable Information
PMBOK®	Project Management Body of Knowledge
PM	Program (or Project) Manager
PMI	Project Management Institute
РМО	Program (or Project) Management Office
POAM	Plan of Action and Milestones
POR	Program of Record
PWS	Performance Work Statement
QA	Quality Assurance
REST	REpresentational State Transfer
RFP	Request for Proposal (Best Value contract)
RFQ	Request for Quotation (Lowest Cost contract)
RMB	Risk Management Board
ROI	Return on Investment



SAML	Security Access Markup Language	
SEI	Software Engineering Institute	
SERB	Software Engineering Review Board	
SIP	System Identification Package (DIACAP artifact required for accreditation)	
SME	Subject Matter Expert	
SOA	Service-oriented Architecture	
SOAR	State of the Art Report (created by IATAC)	
SP	Special Publication (created by NIST)	
SSR	Software Security Review (part of SEI's IMAF methodology)	
STIG	Security Technical Implementation Guide	
STR	Software Test Report (MIL-STD-498 standard artifact)	
U.S.	United States	
WBS	Work Breakdown Structure (project management artifact)	
XML	eXtensible Markup Language (data exchange)	

About the Author

Andrew Bruce is a Lead Executive Scientist for Computer Sciences Corporation (CSC) in the Army Programs group of the North American Public Sector. CSC provides professional services to the Federal Government and the Department of Defense, specializing in customizing and developing architecture and governance models that enable tight integration to the Army's enterprise portfolio management initiatives. Mr. Bruce's job responsibilities include: working directly with customers and partners for new business development, supporting proposal efforts, overseeing Army customers' network infrastructure, working with project managers to ensure project completion, managing software development efforts throughout the entire system life-cycle, and leading new technology research and proofs-of-concept. Mr. Bruce holds the CISSP, PMP, FITSP-D, and ITIL v3 certifications and is currently pursuing a Master's Degree in Information Assurance from Norwich University.



Reference List

The following references contributed authoritatively to this paper.

- [BOS09] Bosworth S, Kabay ME, Whyne E, editors. 2009. Computer Security Handbook. 5th ed. Hoboken (NJ): John Wiley & Sons, Inc. 2035 p.
- [CNSS-1253] Committee on National Security Systems. CNSSI Instruction No. 4009: National Information Assurance (IA) Glossary. April 26, 2009. http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf>. Accessed: May 2, 2011. 103 p.
- [DAG] Department of Defense. August 10, 2010. Defense Acquisition Guidebook [Internet]. http://at.dod.mil/docs/DefenseAcquisitionGuidebook.pdf>. Accessed: May 2, 2011. 929 p.
- [DOD-5000.2] Department of Defense. December 8, 2008. DoD Instruction 5000.02: Operation of the Defense Acquisition System. http://www.dtic.mil/whs/directives/corres/pdf/500002p.pdf>. Accessed: May 2, 2011. 80 p.
- [DOD-8000.1] Department of Defense. February 10, 2009. DoD Directive 8000.01: Management of the Department of Defense Information Enterprise. http://www.dtic.mil/whs/directives/corres/pdf/800001p.pdf>. Accessed: May 2, 2011. 11 p.
- [DOD-8500.1] Department of Defense. October 24, 2002. DoD Directive 8500.01: Information Assurance (IA). http://www.dtic.mil/whs/directives/corres/pdf/850001p.pdf>. Accessed: May 2, 2011. 22 p.
- [DOD-8500.2] Department of Defense. February 6, 2003. DoD Instruction 8500.02: Information Assurance (IA) Implementation. http://www.dtic.mil/whs/directives/corres/pdf/850002p.pdf>. Accessed: May 2, 2011. 102 p.
- [DOD-8510.1] Department of Defense. November 28, 2007. DoD Instruction 8510.01: DoD Information Assurance Certification and Accreditation Process (DIACAP). http://www.dtic.mil/whs/directives/corres/pdf/851001p.pdf>. Accessed: May 2, 2011. 53 p.
- [DOD-8570.1] Department of Defense. December 19, 2005. DoD Instruction 8570.01-M: Information Assurance Workforce Improvement Program. http://www.dtic.mil/whs/directives/corres/pdf/857001m.pdf>. Accessed: May 2, 2011. 97 p.
- [DOD-8580.1] Department of Defense. July 9, 2004. DoD Directive 8580.1: Information Assurance (IA) in the Defense Acquisition System. http://www.dtic.mil/whs/directives/corres/pdf/858001p.pdf>. Accessed: May 2, 2011. 11 p.
- [EGYED] Egyed, A. 2006. Tailoring Software Traceability to Value-Based Needs in Value-Based Software Engineering (VBSE), editors: S. Biffl, A. Aurum, B. Boehm, H. Erdogmus, and P. Grünbacher. Springer Verlag. http://www.alexander-egyed.com/publications/Tailoring_Software_Traceability_to_Value-Based_Needs.pdf>. Accessed: May 2, 2011. p 287-308 (book chapter).
- [MITRE] Northern C, Mayfield K, Benito R, Casagni M. December 15, 2010. Handbook for Implementing Agile in Department of Defense Information Technology Acquisition. MITRE. http://www.mitre.org/work/tech_papers/2011/11_0401/. Accessed: May 2, 2011. 107 p.
- [NDIA08] National Defense Industrial Association (NDIA) System Assurance Committee. 2008. Engineering for System Assurance. Arlington, VA: NDIA. http://www.acq.osd.mil/se/docs/SA-Guidebook-v1-Oct2008.pdf>. Accessed: May 1, 2011. 180 p.



- [NIST 800-50] Wilson J, Hash J. October, 2003. Special Publication 800-50: Building an Information Technology Security Awareness and Training Program. Gaithersburg (MD). http://csrc.nist.gov/publications/nistpubs/800-50/NIST-SP 800-50.pdf>. Accessed: May 14, 2011. 70 p.
- [NIST 800-53] NIST. August, 2009. Special Publication 800-53: Recommended Security Controls for Federal Information Systems and Organizations. Gaithersburg (MD). http://csrc.nist.gov/publications/nistpubs/800-53-Rev3/sp800-53-rev3-final_updated-errata_05-01-2010.pdf>. Accessed: May 2, 2011. 237 p.
- [NIST 800-64] Kissel R, Stine K, Scholl M, Rossman H, Fahlsing J, Gulck J. October, 2008. Special Publication 800-64: Security Considerations in the System Development Life-cycle. Gaithersburg (MD). http://csrc.nist.gov/publications/nistpubs/800-64-Rev2/SP 800-64-Revision2.pdf>. Accessed: May 2, 2011. 67 p.
- [NIST 800-142] Kuhn DR, Kacker RN, Lei Y. October, 2010. Special Publication 800-142: Practical Combinatorial Testing. Gaithersburg (MD). http://csrc.nist.gov/groups/SNS/acts/documents/SP 800-142-101006.pdf>. Accessed: May 2, 2011. 81 p.
- [PMI] Project Management Institute. 2008. A Guide to the Project Management Body of Knowledge (PMBOK[®]). 4th ed. Newtown Square (PA): PMI. 497 p.
- [ROYCE 70] Royce W. 1970. Managing the Development of Large Software Systems. Proceedings of IEEE WESCON 26 (August): 1–9.
- [SEI09] Alberts C, Smith J, Woody C. February 2009. Multi-view Decision Making (MVDM) Workshop. Software Engineering Institute. http://www.cert.org/archive/pdf/08sr035.pdf>. Accessed: April 30, 2011. 36 p.
- [SEI10] Alberts C, Allen J, Stoddard R. September 2010. Integrated Measurement and Analysis Framework for Software Security. Software Engineering Institute. http://www.cert.org/archive/pdf/10tn025.pdf>. Accessed: April 30, 2011. 77 p.
- [SEI11] Alberts C, Allen J, Stoddard R. 2011. Security Measurements and Analysis. Software Engineering Institute. <<u>http://www.cert.org/archive/pdf/SecurityMeasurementandAnalysis.pdf</u>>. Accessed: April 30, 2011. 49 p.
- [SOAR]. Goertzel KM et. al. July 31, 2007. Software Security Assurance. IATAC. Herndon (VA). http://iac.dtic.mil/iatac/download/security.pdf>. Accessed: May 2, 2011. 396 p.
- [STE03] Stephenson PR, ed. 2009. Information Security Essentials (Section 3). Auerbach Publishing, ISBN 978-1-4398-0030-0. 684 p.
- [STE04] Stephenson PR, ed. 2009. Information Security Essentials (Section 4). Auerbach Publishing, ISBN 978-1-4398-0030-0. 454 p.



Miscellaneous References

The following references contributed indirectly to this paper or provide additional information for the interested reader.

Associated Press. March 13, 2009. GAO Highlights DoD Cost Overruns. <<u>http://www.military.com/news/article/gao-highlights-dod-cost-overruns.html</u>>. Accessed: April 25, 2011.

- Bandor MS. September 10, 2009. CDRL Specific DIDs VS Only IEEE Requirements. Acquisition Community Connection. <<u>https://acc.dau.mil/CommunityBrowser.aspx?id=254017</u>>. Accessed: April 28, 2011.
- Benjamin, R. July 22, 2009. Know when to not use agile methodology. http://it.toolbox.com/wiki/index.php/Know_when_to_not_use_agile_methodology. Accessed: May 15, 2011.
- Beck K et. al. 2001. Manifesto for Agile Software Development. <<u>http://agilemanifesto.org/</u>>. Accessed: April 25, 2011.
- Bandor MS. September 10, 2009. CDRL Specific DIDs VS Only IEEE Requirements. Acquisition Community Connection. <<u>https://acc.dau.mil/CommunityBrowser.aspx?id=254017</u>>. Accessed: May 2, 2011.
- Brown University. April 21, 2011. John Savage Testifies Before Senate Committee on Cyber Security. Brown University. <<u>http://sca2002.cs.brown.edu/news/2011/0421.jestestimony.html</u>>. Accessed: May 2, 2011.
- Charles River Analytics. 2004. About Bayesian Belief Networks. Charles River Analytics. <<u>https://www.cra.com/pdf/BNetBuilderBackground.pdf</u>>. Accessed: May 2, 2011.
- Deemer et. al. 2009. The Scrum Primer version 1.2. The Scrum Alliance. http://www.scrumalliance.org/resource_download/339. Accessed: April 25, 2011.
- Gray L. 1999. A Comparison of IEEE/EIA 12207, ISO/EIC 12207, J-STD-016, and MIL-STD-498 for Acquirers and Developers. Abelia Corp. <<u>http://www.abelia.com/docs/122_016.pdf</u>>. Accessed: May 2, 2011.
- Harris C. January 7, 2011. Security Breaches By Contractors Could Lead to DoD Exile. ClearanceJobs.com. <<u>http://tinyurl.com/dod-exile</u>>. Accessed: May 2, 2011.
- Infosecurity. February 18, 2011. Insecure software plays key role in creating cybersecurity vulnerabilities. infosecurity.com. <<u>http://www.infosecurity-magazine.com/view/16066/insecure-software-plays-key-role-in-creating-cybersecurity-vulnerabilities/</u>>. Accessed: April 25, 2011.
- Kundra V. December 9, 2010. 25 Point Implementation Plan To Reform Federal Information Technology Management. <<u>http://tinyurl.com/kundra-25</u>>. Accessed: May 2, 2011.
- Morales, AW. October 1, 2010. Putting some teeth into software security. SD Times. <<u>http://tinyurl.com/sd-security</u>>. Accessed: May 2, 2011.
- NIST. May 21, 2011. Special Publications (800 Series). Computer Security Division / Computer Resource Center. http://csrc.nist.gov/publications/PubsSPs.html. Accessed: May 21, 2011.
- Pronsc M. January 25, 2010. Agile is for Real, says Forrester. AgileZone. <<u>http://agile.dzone.com/articles/45-developers-surveyed-use</u>>. Accessed: April 25, 2011.

RSA Laboratories. 2010. What is a one-way function? EMC Corporation.



<<u>http://www.rsa.com/rsalabs/node.asp?id=2188</u>>. Accessed April 22, 2011.

Sternstein A. March 21, 2011. RSA security breach compromised federal network ID tool. nextgov. <<u>http://www.nextgov.com/nextgov/ng_20110321_6511.php</u>>. Accessed: April 27, 2011.

Wingfield N, Sherr I, Worthen B. April 27, 2011. Hacker Raids Sony Videogame Network. WSJ.com. <<u>http://tinyurl.com/wsj-sony-hack</u>> Accessed: April 27, 2011.