Software Engineering Institute

DoD Information Assurance and Agile: Challenges and Recommendations Gathered Through Interviews with Agile Program Managers and DoD Accreditation Reviewers

Stephany Bellomo
Carol Woody

November 2012

TECHNICAL NOTE
CMU/SEI-2012-TN-024

Acquisition Support Program

http://www.sei.cmu.edu

Carnegie Mellon

This material is based upon work funded and supported by the United States Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

This report was prepared for the

SEI Administrative Agent
ESC/CAA
20 Schilling Circle, Building 1305, 3rd Floor
Hanscom AFB, MA 01731-2125

Table of Contents

List of Figures

List of Tables

Acknowledgements

Special thanks for the contributors to this report:

Lt Col Andrew J. Berry, USAFR; Director, Warfighter's Edge

James L. Boston, Agile Development Team Lead, Jacobs Technology, TYBRIN Corp.

John C. Cargill, Operations Research Analyst, Air Force Cost Analysis Agency

Jeff Davenport, Senior Member of Technical Staff, Software Engineering Institute, Carnegie Mellon University

Edward H. Deets III, Rear Admiral, United States Navy Retired, currently Carnegie Mellon University, Software Engineering Institute, CERT

Christy L. Hermansen, Product Design, Sphere of Influence (https://www.SphereOfInfluence.com)

Joel McAteer, Information Assurance Manager, Modeling Simulation and Instrumentation, U.S. Army Operational Test Command

G. Scott Pringle, Advanced Engineering, Sphere of Influence (http://www.SphereOfInfluence.com)

Donna G. Schutzius, certified information systems security professional, Warfighter's Edge Information Assurance Manager; former U.S. Air Force Academy Certifying Authority Representative and former chief, U.S. Air Force Information Assurance Policy Branch.

Abstract

This paper was produced by the Software Engineering Institute at Carnegie Mellon University in support of the Agile acquisition research agenda funded by the Office of the Secretary of Defense. This paper is part of a larger research study focused on understanding the implications of applying a rapid, incremental development approach, such as Agile, on the Department of Defense (DoD) acquisition process. An overarching goal of this research agenda is to identify areas of tension between Agile and existing processes and provide recommendations for improvement to those processes. In support of the overarching research agenda, several "point" papers are being developed on particular topic areas. The topic of this particular paper is the natural tension between rapid fielding and response to change (characterized as agility) and DoD information assurance policy. The authors gathered information for the paper primarily by conducting interviews with several DoD project managers and information assurance representatives. The interview findings are organized into a list of key challenges and recommendations. The paper also includes a five- to ten-year future outlook with respect to information assurance and agility in DoD. The opinions, findings, conclusions, and recommendations expressed in this Technical Note are those of the authors and do not necessarily reflect the views of the United States Department of Defense.

1 Overview and Scope

This paper was produced by the Software Engineering Institute at Carnegie Mellon University in support of the Agile acquisition research agenda funded by the Office of the Secretary of Defense. This paper is part of a larger research study focused on understanding the implications of applying a rapid, incremental development approach, such as Agile, on the Department of Defense (DoD) acquisition process. The research agenda covers a wide variety of topics throughout the acquisition life cycle (everything from acquisition strategy to requirements and design). The opinions, findings, conclusions, and recommendations expressed in this Technical Note are those of the authors and do not necessarily reflect the views of the United States Department of Defense.

An overarching goal of this research agenda is to identify areas of tension between Agile and existing processes and provide recommendations for improvement that will be implemented over the long term. While long term improvement is critical for meeting the demands for rapid fielding expected of DoD software projects over the next few years, project managers can't necessarily wait for long term process improvement. Therefore, a secondary goal for this research is to synthesize and share the experiences of successful Agile practitioners in the DoD space to improve project manager's chances of success within today's constraints.

In support of this research agenda, several "point" papers are planned which will focus on specific topic areas. The topic of this particular paper is the tension between agility and DoD information assurance (IA) policy. We gathered information for the paper primarily by conducting interviews with several DoD project managers and information assurance representatives. We summarized the interview findings into a list of recommendations. To make sure we are also looking ahead, we include a section on what to expect five to ten years in the future with respect to information assurance and support for agility.

Here is an outline of the remaining paper contents:

- overview of highlights from brief literature search
- summary of recommendations (from all interview findings)
- interview findings introduction
- interview findings from the Agile program manager perspective
- interview findings from the accrediting reviewer perspective
- implications of existing IA processes on Agile principles
- looking ahead: what to expect five to 10 years from now

We close the paper with suggestions for future work in this topic area.

Before we get started, a few quick disclaimers. This paper is not intended to be a formal research report. Although data was collected in a fairly methodical way, no formal research methodology was applied to data collection and analysis. This data collection process is described in Section 2, Research Approach. In addition, the scope of this paper is restricted to the Department of Defense

Information Assurance Certification and Accreditation Process (DIACAP). So, while we acknowledge a close relationship between information assurance accreditation and testing, the scope of this paper does not include challenges related to software testing. However, our intention is to address Agile and software testing in another point paper.

We don't want readers to draw the conclusion that by articulating these challenges we are saying Agile has more information assurance risks than the waterfall life cycle. This is most certainly not the case. The waterfall life cycle has its own set of information assurance-related risks. There are risks and tradeoffs associated with any life cycle.

2 Research Approach

The research approach contains four major pieces. First, we conducted a very brief literature search. Second, we captured challenges and recommendations from interviews with Agile project managers and information assurance representatives. Third, we summarized recommendations. Fourth, we gathered information from a security expert to provide a look forward at information assurance in the DoD space (five to 10 years in the future) derived through expert opinion.

The interview data gathering and analysis method used was informal, but fairly structured and methodical. We maintained consistency in the types of data gathered by structuring the interviews around a generally consistent set of questions (we allowed some tailoring to accommodate different roles of interviewees). After conducting the interviews, we analyzed the data looking for common challenges and recommendations. Where we found common challenges or recommendation themes (e.g., two or more people generally made the same recommendation) we include the results in the paper. If only one person raised the challenge or recommendation, we either dispensed of it or we point out in the paper that there was no corroborating data.

The interviews were conducted with representatives from the following communities:

- program managers for Agile projects (government staff) and Agile software engineers (contractor staff)
- information assurance reviewers or organizational representatives (government staff)

3 Overview of Highlights from Brief Literature Search

As part of this work, we conducted a brief literature search on information related using the following key words and phrases: *Agile and information assurance; Agile and DoD certification and accreditation, Agile testing and certification.* The bullet list below summarizes some of the interesting ideas prevalent in the literature we reviewed:

- Agile appears to be in DoD to stay.
- While DoD certification and accreditation processes don't prohibit the use of Agile, use of them does introduce some challenges related to delivering software features rapidly and/or incrementally.
- Challenges with respect to testing and certification are found in industry, too (e.g., water-Scrum-fall).

The Office of the Secretary of Defense and other organizations have released in the public domain several documents indicating that Agile is not a passing fad in DoD; rather Agile is here to stay [DoD 2009, DoD 2010, NDAA 2009, NDAA 2010, Section 804 2012, DoD 2010a]. Many of the directives strongly suggest DoD information technology (IT) projects follow development processes and life cycles patterned after Agile [DoD 2010a]. In addition, several supporting federal government and DoD task forces and working group initiatives are underway to support the implementation of these directives, such as the Section 804 Task Force [AFEI 2012]. On reading the directives, it is clear the government is banking on alternative development processes to deliver capability to the user sooner. Therefore, it would be prudent for government program managers, as well as accreditation authorities, to start thinking about the implications of applying these methods on their own projects as early as possible.

As part of the literature search, we briefly reviewed current DIACAP information (such as the diagram below). There is nothing in DIACAP that prevents project teams from developing software incrementally. The short green bars shown in Figure 1 indicate that incremental development is supported by DIACAP. However, DIACAP Activity 3 ultimately serves as a gating function for operational release of capability into the field. A favorable certification determination depends on verification that all required security controls are implemented and certified prior to operational release.

While in general security controls must be addressed prior to operational release (especially for systems that include hardware and software changes), it is important to point out that DIACAP supports a course-grained structure for prioritization of security requirements. DIACAP contains a vulnerabilities rating scale where vulnerabilities are assigned to category (CAT) 1, 2 or 3. Each category maps to a severity level. For example, DIACAP directives and policy will not allow for a program or an application with CAT 1 vulnerability to be issued an approval to operate unless the highest level authority in the information assurance approval chain approves it. If a project team has not met a required security control they can address this in DIACAP Activity 2. The project team must add the control to the plan of action and milestones (POA&M) which contains a list of

all the IA controls that are not mitigated. Using this information, a designated approving authority (DAA) or a certifying authority can make a risk assessment decision.

In a nutshell, we found that while DIACAP doesn't prohibit the use of incremental life cycles such as Agile, there are aspects of DIACAP that make incremental delivery of security requirements challenging. Some of these aspects are related to the high-risk nature of security-related requirements while some aspects are related to the nature of the review process itself. For example, DIACAP accommodates iterative development; however, the gating nature of the accreditation process creates a bit of a waterfall life cycle [DIACAP 2012]. The "big bang" approach to accreditation review often results in many security requirements being evaluated at the end of development.[1]

The pressure on Agile projects to demonstrate business functionality early combined with a lack of emphasis on security requirements early in the life cycle often causes problems. The information assurance review step becomes flooded with security requirements resulting in release bottlenecks and schedule delays. There were many interesting perspectives that surfaced during the interviews. Several of the interviewees felt that the information assurance accreditation delay is so extensive (often months to a year) that the DIACAP process almost negates the benefits gained through rapid development methods. However, one interviewee provided an alternative viewpoint on this topic also shared by several others. They said, "The problem is that most [Agile] programs think that the end user is their only customer." This is false. The information assurance stakeholder is one of many other customers that must have input into the backlog prioritization process."

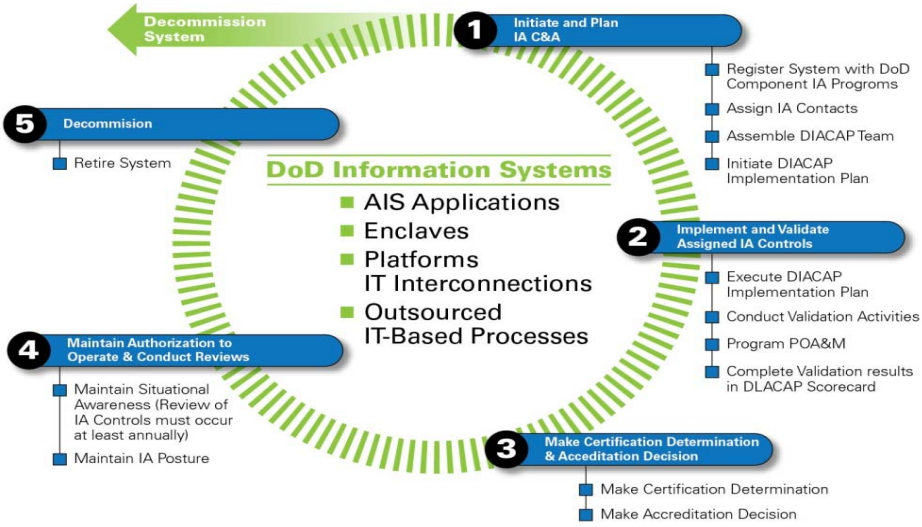


Figure 1: Department of Defense (DoD) Information Assurance Certification and Accreditation Process (http://www.diacap.org)

A significant part of the DIACAP is related to addressing the IA controls prior to issuance of authority to operate (ATO). DoD program managers are required to comply with the DoDI 8500-2 IA control requirements (step 4 in the figure above) [DoDI 2008]. However, some organizations

[1] Expert opinion of Jeffrey Davenport, Software Engineering Institute, Carnegie Mellon University

are starting to recognize the need for more flexibility. For example, one interviewee noted that the Army has a standalone information security and closed restricted networks information assurance certification and accreditation (IA C&A) requirements process that drastically reduces the number of IA controls for certain types of systems. This minimizes allocation of time and resources against low priority requirements or requirements that are not relevant for the environment or context [Agile 2011].

Another interviewee said that the Air Force has incorporated some mechanisms for flexibility into the process. One interviewee said that the Air Force does not force software applications to go through DIACAP. It modified the process so that only systems with hardware and software go through DIACAP. Software "only" changes go through the software certification process administered by the Air Force Network Integration Center (AFNIC). The AFNIC commander has been designated by the Air Force DAA as the approving official for software going on the .af.mil portion of the global information grid (GIG).

During the literature search we also found that testing and certification delays for Agile projects are not limited to DoD. There is wide recognition in industry that there is a need to better support streamlined test and certification processes for Agile projects. In our literature search we came upon a heavily referenced Forrester report titled *Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today*. This paper served as the inspiration for many related blog postings, online community discussions and related reports. The crux of the paper is described in the report's executive summary, which states that Agile adoption has diverged some from the original ideas described in the Agile Manifesto. Many adoptions resemble what Forrester labels "water-Scrum-fall." The "fall" part of water-Scrum-fall reflects the inherent challenge in trying to get organization-wide testing and certification (and governance) processes and policies to align with the business goal of rapidly fielding capability. The report explains that the water-Scrum-fall model is not necessarily bad or avoidable; however, if application development professionals do not make the right decisions about where the lines fall between water-Scrum-fall they are unlikely to realize all of the benefits of Agile.

The Forrester paper supports the notion that organizations in industry, like DoD, struggle to balance security and testing policy with the desire to deliver features sooner [West 2011]. Consequently, useful knowledge could be gained by maintaining awareness of how industry is responding to these types of challenges.

4 Summary of Recommendations from Interviews

The section below contains combined summarized recommendations from the interviews with Agile program managers (and teams) and accrediting authorities. The elaborated interview findings can be found in Sections 5, 6 and 7 of this report.

Recommendation 1: Define criteria for reaccreditation early in the project.
Streamlining of the accreditation process is greatly needed; therefore, many of the subsequent recommendations in this report focus on the topic of streamlining the initial certification phase. However, several interviewees also acknowledge that changing a DoD process, such as a certification process, is a lengthy endeavor. The problem is that even if process changes are incorporated into the process, these may not be here in time to help project teams deal with challenges today. Therefore, a multi-pronged approach is needed. Several interviewees suggested that in addition to streamlining the initial certification process, there is great value in focusing attention on the reaccreditation process. Reaccreditation occurs after an organization has received approval to operate. The general concept is, rather than going through full reaccreditation for minor modifications which may take months or longer, the project team need only to reaccredit the portions of the architecture that have changed (assuming they can agree on these rules up front). This is an area where Agile project teams have an opportunity to influence the process since much of the process is defined jointly by the project team and accrediting authority. To support an effective reaccreditation process, several interviewees recommended that Agile project teams work with accreditation reviewers to define common criteria for reaccreditation as early as possible in the project lifecycle. For example, interviewees suggest it is critical to have agreement is regarding what constitutes a major versus minor release (e.g., certain types of software changes, hardware changes, or both).

One interviewee provided a verbal description of the successful reaccreditation process they use, illustrated below. Please note that the illustration is a conceptual representation of the example provided verbally during this interview and is not intended to be a formal representation of the described process.

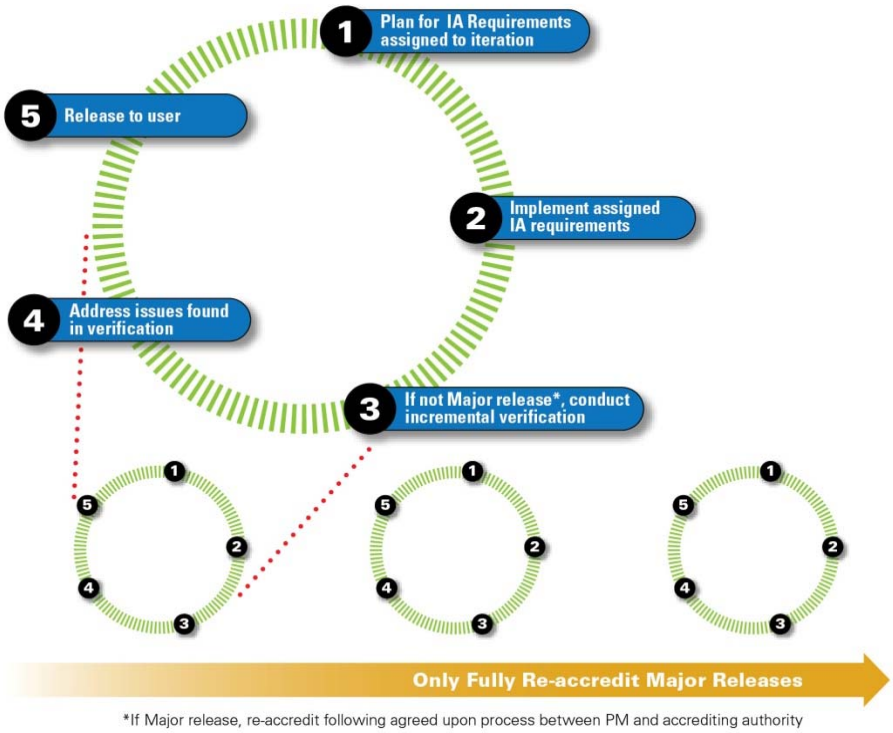


Figure 2: Reaccreditation Process Described in Interview

The next several paragraphs describe the example illustrated in the figure above. In this example the baseline accreditation is complete and a new increment is being planned. Therefore, the process describes their reaccreditation process. The steps of the reaccreditation process as illustrated are outlined below:

- plan for IA requirements to be assigned to next iteration
- implement planned IA requirements for iteration
- conduct incremental verification (if not major release)
- address issues found in verification
- release to user

This paragraph describes the steps above in greater detail. In the first step, the project team works with the accrediting authority to plan for upcoming increment(s). Collaboratively, the team prioritizes the requirements and assigns them to an increment. At this point, an initial determination is made as to whether this is expected to be a “major” or “minor” release. In the second step, the Agile project team implements the requirements assigned to the increment. In the third step, if the release is formally determined to be a major release as per the agreed upon reaccreditation criteria, the relevant artifacts are verified by the accrediting authority. In the fourth step, the Agile project team addresses issues found during the verification step. Finally, upon successful verification the iteration is released to the user. In this example, the key differentiator between a minor release and a major release is impact to hardware platform or interfaces. If the changes are determined to

be software-only changes, the release is generally determined to be a minor release. A minor release has minimal, if any, reaccreditation review requirements.

A different interviewee provided another example of reaccreditation release criteria, shown in Table 1. Like the previous example, the criteria below are based on the impact of the change. Notice that the minor release reaccreditation requirements list is much shorter.

Table 1: Reaccreditation Release Criteria

Major Release Reaccreditation Criteria[2]	Minor Release Reaccreditation Criteria
• Review all architectural documentation • General functionality test • Test any change in the security design (breaks the defined security criteria) • Test public interfaces • Test authentication model • Test encryption • Test services	• Localized review based on change • Review of documentation related to localized changes • Testing as needed to test change

Leverage long accreditation approval wait time with frequent community previews.
Many of the interviewees complained slow accreditation approval process impacts their ability to rapidly deliver new capability into the field. To work around this, several interviewees suggested making productive use of the time waiting for accreditation approval by conducting user feedback demos (referred to by most interviewees as community previews). This shortens the release time in the long run because user acceptance testing generally goes more smoothly if users have been providing feedback and it has been incorporated prior to production release.

Ensure requirements prioritization of backlog considers business value and risk.
Because of all the hype around Agile projects' early delivery capability, it is easy to slip into a pattern of prioritizing based on business value alone and not risk. If unchecked, this can lead to pushing off critical security requirements to the end of the development cycle where problems can be very costly and impactful. To address this, several interviewees recommended that Agile project teams include stakeholders that represent the security perspective into the requirements prioritization process early. One interviewee representing the information assurance community provided this comment to emphasize the importance DoD places on security requirements (from the *Security and Development Security Technical Implementation Guide,* or *STIG*): "No software should be presented to DoD with a CATEGORY 1 vulnerability as defined by the DoD STIG nor should software products use any prohibited (i.e., red) ports, protocols or services as defined by DoD Category Assurance List" [DISA 2011].

Make sure Agile project teams understand the intent behind security requirements and organize backlog accordingly.
Some interviewees observed that project teams appear to be working from a list of controls rather than documentation that describes the intent behind the control. Without the information describing the intent behind the control, it is difficult for engineers to do risk tradeoff analysis and reason

[2] Please note: The information in this table is provided to convey the concept informally described by the interviewee who provided the example and is not intended to reflect a complete listing of the criteria.

about backlog prioritization with information assurance authorities. Representatives from the information assurance community we interviewed disagreed with the assertion that intent is not covered in the control documentation and provided documentation to counter this assertion. For example, one interviewee suggested that all Agile teams read the DISA *Security Technical Implementation Guide* prior to writing code [DISA 2011]. Regardless, the program manager or Scrum master must make sure that the project engineers are equipped with the information they need, including information about intent behind controls.

Ensure Agile development processes produce and maintain "just enough" design documentation.
Agile development processes don't necessarily produce a significant amount of documentation as a natural by-product of the Agile development process. To augment security risk analysis needed to support accreditation, there may need to be a step interjected into the Agile process to produce verifiable evidence necessary to reason about risks and vulnerabilities (but not too much as to be wasted effort). During the project initiation phase when developing the runway [Leffingwell 2011] it is necessary to identify the minimal set of security-related architecture design documentation needed to reason about design of high criticality security features. This may include relevant design views, description of key components, rationale for design decisions, variability mechanisms, and the like [Clements 2011]. The security architecture design documentation also needs to be maintained throughout the project. During the maintenance phase, this design documentation will be leveraged to assess impact as changes are introduced. One interview candidate described a successful tactic they use to ensure that risk analysis is part of the change process. They use a product backlog item (PBI) template that includes a security impact assessment field. Enforced use of this field increases the likelihood that security risk is continuously assessed by the team as new features are introduced.

Make sure there is at least one person with strong security analysis expertise on the Agile project team. In order to reason about incrementally delivering security features or even to deliver changes through a rapid release life cycle (e.g., four-week release cycle) the project team needs to have least one person who can reason and articulate the risk related to a security requirement. This is of critical importance if the Agile team wants to move toward prioritizing security requirements as part of the backlog-driven requirements process. The Agile project team must ensure that someone is clearly assigned the responsibility for risk analysis. To maintain credibility with the accreditation organization, it is important that project teams not push risk analysis responsibility onto the accreditation authority. One interviewee said that the responsibility for security risk analysis is assigned to the chief architect on their team. This chief architect oversees the design for multiple teams. Another interviewee said risk analysis is a shared responsibility across teams. Regardless of approach, whoever is assigned responsibly designing and maintaining the security design also needs to be capable of articulating risk related to deferring a security requirement and developing a mitigation plan for all unaddressed risks. One interviewee suggested that the chief architect can't do risk analysis without strong design documentation (mentioned above). The security design is needed to make decisions about requests for security requirement waivers and to assess associated risk related to delaying requirements.

Foster Agile project team and accrediting authority collaboration.
Agile promotes a collaborative development and testing philosophy. While it is important to maintain independence between Agile project teams and accreditation reviewers for security accreditation, there is also value in having close collaboration with information assurance experts. Several interviewees representing both the Agile team and accreditation authority perspectives suggested embedding an information assurance consultant on the project team to help with the security design. One interviewee emphasized the importance of having the collaborative information assurance on the team, as described in the quote below.

> ***"The trick is to find creative and collaborative people with IA expertise to help Agile projects understand risks but still keep the project moving in a forward direction."***
> ***—DoD accrediting authority***

One interviewee suggested that one benefit to having an embedded information assurance consultant on the team, at least in the early stages of development, is that they can guide and train the developers to build their security expertise. In one interesting anecdote told by an interviewee, a project lead tested their development team's security skills by having the team go through certified hacker training. After the training, the project team conducted an exercise by simulating an attack on the system. In this type of activity having an onsite information assurance consultant is of great value to the project team.

Don't apply all the information assurance controls blindly.
Several interviewees, including accreditation reviewers, said that if all the information assurance controls were implemented most systems would not be able to operate because they would be so locked down. In a blog posting on http://www.sectool.org (a 3,000 member shared security community) one person expressed this word of caution regarding accepting all the vulnerability fixes found in a scan wholesale: "The DISA Gold Disk will scan your system in two to three minutes and identify many vulnerabilities," the person wrote. "If you blindly apply all of the recommended fixes, it will also break your system very fast, guaranteed..." [Security 2005]. While it is important to address the required controls in your design, is not good to apply all the information assurance controls blindly. Instead review the controls to understand design implications. If a control does not seem appropriate for your situation, it may be necessary make a request to the accrediting authorities to accept the risk. One of the interviewees suggested that this should be what the DIACAP implementation plan (DIP) is for, but they added that unfortunately it is not used this way in practice. The information assurance consultant can be instrumental in helping the team analyze and articulate risk tradeoffs to information assurance authorities.

Use common operating environment (COE), software development toolkits (SDKs) and enterprise services to speed up accreditation time.
Several interviewees suggested that their Agile project teams have reduced accreditation time by leveraging COEs, SDKs, and enterprise services. The idea is that the use of well-vetted, precertified platforms and standards reduces security risk to some degree and, therefore, speeds up the accreditation review process. The use of COEs and SDKs is particularly prevalent in the Army. The Army Software Transformation Common Operating Environment concept is described

on the Army website as "a set of computing technologies and standards that will enable secure and interoperable applications to be rapidly developed and executed across a variety of computing environments: server, client, mobile devices, sensors, and platforms…. to deliver relevant applications to those who need them…" [Army 2010]. In addition, mobile app submission to the "Apps for the Army" contest requires the use of approved SDKs. In the future it is anticipated that these SDKs may provide security policy checking capability so that some of the security requirements are automatically enforced prior to submission. One interviewee described using approved services provided by the enterprise, such as role-based access services, to speed up development and accreditation for that feature set. Agile program managers and teams should take advantage of these and other certified resources to speed up the accreditation process.

This recommendation is not to be taken lightly. Another interviewee spoke of the security implications related to the move several DoD enterprises are making toward platform consolidation. The interviewee shared this observation with us: "We have also seen a lot of consolidation of programs going on as budgets get tighter. We have a big push to shift to consolidated environment and enterprise services that can support a wide range of users and capabilities across programs. This has security implications as well as affecting how we view the promise of Agile development. Taking a holistic view of how a capability can be leveraged across tens of systems or more is challenging. The question is how does Agile fit into that kind of future?"

Apply a risk-based, incremental approach to security architecture.
The Agile design process is typically more of an evolutionary design process than the traditional design-it-all-up-front process. The problem is that with all of the pressure on DoD programs to deliver capability sooner, it is tempting to focus on developing the low hanging fruit first (the easy stuff) and ignore developing the more complex, high risk capabilities. This can have disastrous results. One interviewee told a story of discovering the need to generate encryption keys for certification of a satellite system very late in the development life cycle causing a significant schedule impact. The interviewee said:

> ***"You can't fire up the equipment if you don't know what keys you can you use. What do you do? Do you use dummy keys? It was a nightmare to determine what could and couldn't be done. Good example of a big risk item pushed off to the end which is what you do not want to do."***
> ***—DoD senior program manager***

The recommendation here is to focus on high risk components and interfaces first. At its core, this recommendation is modern day application of the spiral development approach defined by Barry Boehm in the 1990s [Boehm 1988]. One of the interviewees elaborated by saying, "We always do the more complex features first for each release. We adjust scope throughout—particularly the scope of complex features to leave us time at the end for low hanging fruit. Once we have complex features to an acceptable state, we prioritize all of the low hanging fruit—from the new and existing features. You have to do the hard stuff first or you get yourself in big trouble."

Leverage design tactics such as layering and encapsulation to limit impact of change.
One of the interviewees, with a great deal of expertise in space and large-scale weapon systems, strongly suggested leveraging architectural patterns and tactics such as layering and interfaces to localize change. These patterns and tactics can help limit ripple effects from change and reduce reaccreditation time by localizing change, consequently minimizing what needs to be re-accredited when there is a change. These are just a few of the patterns and tactics that can be used to encapsulate areas anticipated to change. Other patterns and tactics should be applied as appropriate to achieve required quality attributes [Bass 2003]. One interviewee said, "This is good practice under any circumstances. There are also ways to deploy patches without triggering a retest, making deployment a lot faster. Things like changes to security and COTS upgrades automatically trigger a retest. The more you can architect your system to support deployments that don't trigger a retest, the better off you are."

Leverage unclassified environments for Agile development and community previews.
Most of the interviewees we spoke with are developing systems that operate in highly secure environments. While the operational environment for the systems is classified, interviewees recommended that the development be done in the unclassified environment. There are a couple of benefits to using an unclassified environment for development. First, it is easier to do community previews because people can easily get access to the system. Second, having the developers work in the unclassified environment helps to buffer Agile teams from the distractions related to the accreditation process and the limitations of the classified environment. This allows the majority of the developers, with the exception of the chief architect, to focus on maintaining the sprint cadence rather than the ins and outs of the sometimes frustrating and time-consuming accreditation process.

Merge Agile and security best practices (e.g., integrate vulnerability scans into continuous integration process, leverage automated test cases for accreditation validation, adhere to secure coding standards).
The interviewees provided examples of where best practices from Agile and security domains may be merged to shorten accreditation timelines. From the Agile side, some interviewees said their teams were successful integrating vulnerability scans into their continuous integration cycle. This ensures that the scans are constantly being run so newly introduced vulnerabilities are caught early. Interviewees also suggested leveraging automated test cases for accreditation evidence (as opposed to relying so heavily on paper-based review) where approved to do so. From the security side, interviewees recommended that Agile projects ensure team members follow good security practices. For example, project teams should be familiar with ISO 17799 and OASIS Application Security Standards [Coverpages 2012] and security coding standards [CERT 2012]. Some interviewees felt this was so important they suggested creating certification processes for secure coding developers to ensure developers have strong security competency. A good resource for building security-related skills is the Department of Homeland Security Build Security In (BSI) expanding body of knowledge [US-CERT 2012]

Agile and the information assurance community must join forces to continue improving information assurance processes.
Because the focus of the interviews was on providing useful recommendations for dealing with

today’s problems, most of the recommendations assume that we must live within the today’s processes and policy constraints. One interviewee pointed out that there is wide recognition across industry that there is a need to better support Agile projects by improving test and certification processes. Despite the tension between the need for standardization and rapid delivery, the Agile and information assurance communities (contractors and government) need to work together to creatively make these processes better.

5 Interview Findings: Introduction

In the following several sections we present the interview findings from two different perspectives. We present challenges captured through interviews with Agile program managers (and teams) and challenges described by members of the DoD accreditation authority community. As introductory material, here we provide a short overview of Agile practices used on their projects followed by some common themes. These common themes are shared perspectives we heard repeatedly from both the Agile project side and the information assurance reviewers side.

When asked what Agile practices the DoD project teams we interviewed commonly applied on their projects they responded:

- backlog-driven requirements process
- continuous integration
- continuous test
- Scrum management style
- retrospectives
- user demos
- some automated testing

Some of the common themes shared by Agile project managers and information assurance representatives include the following:

1. When asked the overarching question, "Is Agile beneficial to security?" most interviewees agreed it is beneficial (or at least does not make existing security problems worse). Some said shorter iterations make problems visible sooner so problems can be fixed earlier. Some also said shorter increments have the added benefit of providing opportunities to address new vulnerabilities as they emerge. One interviewee said the following:

   > ***"Say you are developing a common operational picture which fields on a five-year cycle. If you find [a] vulnerability, you don't want to wait five years to fix it! You want to fix it as soon as you can—so an incremental life cycle will better support that."***
   > ***— Former Department of Defense accrediting authority***

2. Interviewees strongly support the notion of having the accrediting authorities support two roles: a consulting role where the information assurance expert is embedded in the Agile team, as well as a separate independent reviewer role. These are preferably supported by different people to avoid conflict of interest. Interviewees particularly emphasized the importance of having the information assurance consultant serve as a contributing member of the Agile team early. During the interviews, we heard several examples where this approach was applied successfully. By having both roles supported, project teams have the benefit of

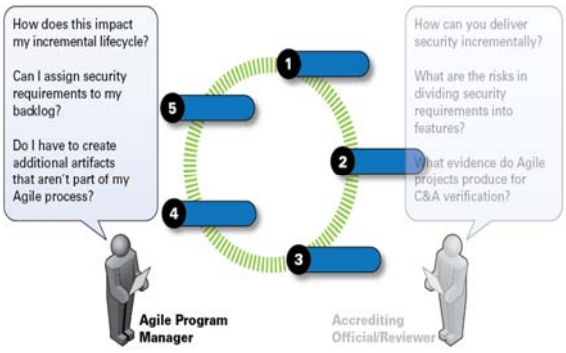
having information assurance expertise on hand to help design security in and project managers can feel comfortable that information assurance reviewer objectivity has been maintained.

3. Many of these issues are caused by assigning security requirements low business value. Another key theme derived from interviewees is that the information assurance stakeholder should be considered a key business stakeholder in the Agile requirements prioritization process. This means that he or she should hold the same status as the user or product owner. Involving information assurance stakeholders in the requirements process ensures security requirements are given appropriate value in the backlog. This may push development of these requirements to earlier increments. The information assurance stakeholder should also participate in subsequent increment planning cycles to ensure that emerging vulnerabilities are assigned appropriate value.
4. Another major theme shared by all the interviewees is the belief that it is critical to define reaccreditation criteria as early as possible in an Agile project. Agile projects anticipate change. Therefore, it is important to have good processes in place to streamline reaccreditation for new increments rather than have to go through full accreditation with each release of functionality. An efficient reaccreditation process requires work in several areas. Criteria for reaccreditation must be established and well-defined. Systems need to be architected to localize change. Agreements need to be in place specifying how the criteria shall be applied (e.g., hardware or software-only releases). Several interviewees provided examples of reaccreditation criteria they use which are included in this paper, Section 4.
5. Many interviewees also noted there is a lack of time or money provided to DoD software projects to build security in from the beginning. In addition, there is a lack of funding for projects already in the software maintenance phase to start emphasizing security if vulnerabilities are discovered in production use. This challenge exists for Agile and waterfall projects.

In the sections that follow, we organize information captured through interviews into three sections:

- interviews with Agile program manager/project team
- interview with information assurance accrediting authority/reviewer
- recommendations derived from combined interviews with Agile project teams and accrediting authorities

6 Interview Findings: Challenges from the Agile Program Manager Perspective



This section describes the challenges, captured through interviews, that Agile project teams encountered as they went through the current DoD certification and accreditation process. (Note that future process challenges are discussed in Section 9). The small pool of interviewees we talked to included program managers that have Agile projects in their portfolio (government personnel) and Agile project engineers (contractor personnel supporting government projects). Where multiple interviewees raised the same challenge, the data point is included below as a challenge or recommendation. If a challenge was raised by a single interviewee, it is either considered a "one off" and is not included or is noted in the text.

Agile projects suffer release delays due to a long accreditation testing phase.
This challenge is not a surprise. Interviewees unanimously complained about the extremely long duration of the security certification and accreditation testing activities on their projects. Several said that this significantly reduces the opportunity to deliver capability to the field quickly. They commonly described a situation where they rush to get a sprint or increment through user acceptance testing and then wait months for security accreditation and approval to field the capability. Some of the interviewees have started to accept the long delays as a fact of life. Some are even happy with waiting months for accreditation as compared to waiting a year or longer, as described in the quote below.

> ***"The process used to be nearly a year long just to get the delivery out but it is getting better. It still takes longer than it should. It is often as much as three months before a release hits operations."***
> ***—DoD Agile program manager (government staff)***

Organizations such as the Army are trying to address this with process adaptions for special cases. The Amy supports DoDI 8510.01 (DIACAP) which allows for an interim authorization to test (IATT). The IATT states that "IATT accreditation decision is a special case for authorizing testing in an operational information environment or with live data for a specified time period. IATTs should be granted only when operational environment/live data is required to complete specific test objectives." IATT skips the independent validator step, saving time. While this is helpful in some cases, interviewees said that at times there is confusion over how to correctly implement the IATT approach.

Regardless of the process followed, waiting three months for certification and accreditation makes attaining DoD goals of releasing every six months [Bellomo 2011] unattainable except for software-only maintenance releases with very minimal localized changes. Generally the interviewees shared the concern that delivering increments early only makes a difference if the changes get deployed to the user in a timely manner. When asked what causes the delays, most of the interviewees suggested that the long certification time is primarily due to the manual nature of the review process and limited certification reviewer resources.

Some also attributed this to late involvement of information assurance stakeholders in the requirements prioritization process. One interviewee shared this observation: "The delays are due to these issues: (1) Most customers do not understand IA, thus accept products they should not (most customers will say they want a secure product and expect the development group to give it to them. The user has no way to verify that the product is secure. Thus IA acts on their behalf). (2) Developers see IA as an afterthought and [don't] recognize them as one of their customers, but as an obstacle to overcome. (The developers are violating the Agile principle 'Business people and developers must work together daily throughout the project.' They are waiting until the end.")

Risk related to security requirements influences how they are prioritized in the backlog. Interview candidates generally agreed that security features need to be given special consideration in the Agile backlog due to the risk related to a security vulnerability. With DIACAP this translates to a requirement that all information assurance controls be addressed prior to deployment. DoD will not authorize the issuance of an authority to operate (ATO) for software delivery with an unmitigated CAT 1 vulnerability except under extreme and rare circumstances. There were differing views on the "all or nothing" DIACAP approach to security controls. One interview candidate suggested that, in his opinion, the DIACAP approach runs counter to the Agile position that time boxes remain fixed and requirements backlog should be reprioritized, based on value and risk, at each increment. This is not really possible with DIACAP. However, DIACAP does provide a formal mechanism for delivering a system without a required control. Unmitigated controls can be included in the POA&M and these vulnerabilities are then assessed on a case-by-case basis by the accrediting authority. A decision to approve a waiver is then made based on risk and strength of the proposed mitigation strategy. So, while security requirements are not like other Agile backlog items, POA&M-based mechanisms do allow for some flexibility if a control is determined to be inappropriate for a particular implementation.

Deficiencies in security engineering analysis expertise limits the Agile project team's ability to do risk tradeoff analysis at the feature level. Several interviewees said a key inhibitor to considering prioritizing security requirements is that DoD software projects struggle to find people with strong enough security experience to reason about the impact of delaying a security requirement or to articulate the importance of addressing a particular vulnerability. For example, a lead project engineer may need to explain to the C&A authorities why a particular CAT 3 vulnerability in a software product is not as critical as a CAT 1 vulnerability. For release planning purposes, it is also necessary for lead engineers to have the design and analysis skills necessary to understand dependencies between the security requirements and the software modules [Brown 2010].

Limited familiarity with intent behind information assurance control.
Another challenge some interviewees observed is that project teams appear to be working from a

list of controls rather than understanding the intent behind the control. While an understanding of the intent is inarguably critical for risk tradeoff analysis it probably even more so in an Agile context where capabilities may be delivered incrementally because it is important to focus on the highest priority risks first. While there was an agreement among interviewees that limited familiarity with control intent is an issue, there were dissenting opinions about where the fault lies. The project management side complained that the control documentation lacks intent information. Representatives from the information assurance community disagreed with the assertion that intent is not covered in the control documentation and provided documentation to counter this assertion. For example, one interviewee provided the DISA *Security Technical Implementation Guide* which does contain examples describing the intent behind controls—at least at a cursory level [DISA 2011]. It is difficult to determine the root cause. Information assurance guidance material may not be as easy to find as project teams need or information may be getting lost as it is passed down from management. Perhaps the project teams do not try hard enough to get this information from information assurance authorities. Regardless, it is up to the program manager or Scrum master to manage this risk and equip developers with the information they need to understand controls and intent.

Lack of clarity over reaccreditation process and reaccreditation criteria.
Another challenge voiced by several interviewees is that after a system is accredited and deployed the process for reaccreditation of subsequent releases is vague and confusing. If this is not managed proactively, a project can end up going through a several-month-long reaccreditation of the whole system for minor changes to the software. To address this, many project teams are leveraging an incremental approach to IA accreditation. At the National Defense Industrial Association (NDIA) C4ISR Agile workshop in November 2011, a keynote speaker from the National Security Agency [NDIA 2011] described a process she called delta accreditation. Rather than going through a full reaccreditation when a project team is making minor, software-only modifications, project teams are only required to go through certification and accreditation for the portions of the system that have changed.

One interviewee said the biggest challenge with reaccreditation is gathering the right people together to develop the reaccreditation criteria early in the process, as described in this quote:

> ***"When it comes to defining reaccreditation release criteria (what goes in to a major versus minor release), you have to get everyone with veto power involved early in the process!"***
> ***—DoD Agile project team member***

The criteria for major and minor changes within DoD services is not universal (even within a particular service) and must be defined jointly by each project team and the accrediting authority for the organization. The same interviewee quoted above cautioned further that the reaccreditation criteria need to be articulated in great detail and suggested it greatly helps to have an accreditation expert working side-by-side with the project team on this.

Limitations in using test case-driven verification approaches for IA validation.
Many of the interviewees said they use Agile test-driven development practices. Some of them

create security stories and then they test the story implementation using automated scripts. These test scripts could potentially speed up the accreditation review process if they could be leveraged. However, test case-based evidence is not generally accepted by certification and accreditation authorities for security requirement validation. Instead certification reviews rely heavily on manually intensive paperwork review and vulnerability scanning software. One interviewee described a scenario where the team had developed scripts for testing role-based access. However, the accreditation reviewers would not allow the scripts to be used in the certification and testing environment because the test scripts had not been accredited—a Catch-22 situation. As a result, the role-based access configuration was tested manually by accreditation authorities adding a great deal of time to the accreditation process.

Agile processes don't naturally produce views required by C&A reviewers.
Several interviewees explained the accreditation community often requests DoD Architecture Framework (DoDAF) views as part of the review process. However, while Agile projects may generate some architectural and security documentation as part of their design process, these views are not typically DoDAF-formatted views. Consequently, Agile teams end up producing additional documentation to meet the review requirements that may not be useful to the team. Some interviewees suggested that perhaps a minimal set of architectural views naturally produced as a by-product of the Agile development process, plus test case-driven evidence, would be more aligned with Agile processes.

Accrediting authority gate-keeping role can impact collaboration.
A small number of the interviewees raised an issue related to the independent nature of the certification and accreditation process. While all interviewees were in agreement that a certain degree of verification independence is necessary given the national security implications for DoD systems, when the role of the accrediting authority becomes combative this limits collaboration. Many of the interviewees said that they have had success when the accrediting authority provides two types of support: as consultants and reviewers. Several interviewees gave examples of security consultants from the organization's information assurance team embedded for a time with their project teams. This helped the project teams develop a solid security design. In most of these examples, a different person served the accreditation reviewer role. The embedded security consultant role is very much in keeping with the Agile philosophy of collaboration—one of the key Agile principles [Agile 2011]—which promotes face-to-face collaboration across all project roles to include developers, testers, and users. One interviewee pointed out that having an accreditation consultant serving as a member of the team also had the additional benefit of enabling rapid decision making.

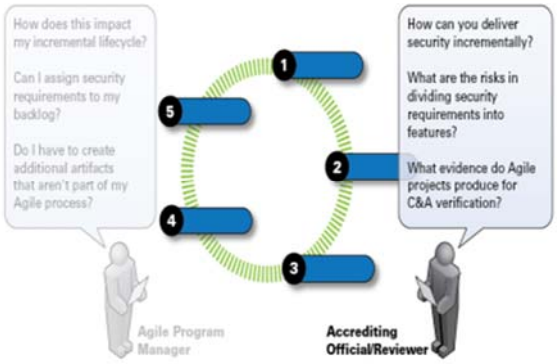
Lack of incentive to develop security features in early increments.
Some interviewees suggested that when security requirements are verified in one fell swoop before operational release there is often little incentive to implement and test security requirements in early increments. The problem is made worse by the fact that requirement backlog is often prioritized by business value alone as opposed to business value and risk. As a result, security features that may have serious design implications for the system are often discovered very late. One interviewee elaborated by saying that when this occurs, proper identification of relevant stakeholders has not occurred early enough in the process. They suggested that security stakeholders

are key stakeholders due to the risk of exposure from unmitigated vulnerabilities and, therefore, should be part of the prioritization process.

Traditional milestone outputs (e.g., Critical Design Review) required to pass accreditation don't map to the Agile process.
Another challenge mentioned by one interviewee is the requirement to provide evidence that the product has successfully passed through some of the traditional milestone gates that are not part of the Agile process (e.g., critical design review, preliminary design review). This required the Agile project team to map traditional milestone outputs to their Agile process outputs. This mapping activity could be considered by some Agile diehards to be "waste" which most Agile methods seek to minimize. Some examples of how this mapping has been done by other programs and the implications of doing this can be found in the paper *Considerations for Using Agile in DoD Acquisition* [Lapham 2010]. Needless to say, this has potential to further bog down the Agile rapid release life cycle, especially if milestone evidence is required for accreditation of major and minor releases.

Implications of DoD Joint Services accreditation processes on Agile projects.
One interviewee raised challenges with processes for joint services accreditation. Although only one person raised this issue, it is include here because of the increasing prevalence of joint military service system initiatives and the impact these types of issues may have on rapid capability delivery. Joint services projects are informally defined here as projects where different military services (e.g., Air Force, Army, Navy) independently develop parts of a joint system. While reciprocity does exist, the processes for joint services accreditation are poorly defined. If a reciprocity agreement is not supported by both parties individual certification may be required for each system and then again for the combined system. One interviewee gave an example to illustrate the problem. They described a joint situational awareness application with data services developed by two different military services. The Agile team's project had completed its portion of the system on time and received accreditation. However, the other service did not successfully complete the accreditation process required by their military service. This caused the operational deployment to be delayed for months. Vulnerabilities discovered late in the life cycle affected both systems. To avoid situations like this one, Agile project teams should work to identify and address joint services risks, and other governance risks, as early as possible.

7 Interview Findings: Challenges from the Accrediting Authority Perspective



In the last section we talked about challenges Agile program managers and project teams face when going through DoD certification and accreditation process. This section addresses the other side of the coin. In this section, we view the accreditation process for Agile projects from the accrediting authority perspective.

Please note that some of these challenges may seem to be covering some of the same ground listed in the prior section. This is because some of the challenges impacting Agile project teams and accrediting authorities stem from common root cause issues that affect both sides.

Security requirements and stories are assigned low priority by stakeholders causing Agile teams to "bolt on security at the end."

Several interviewees described scenarios where security requirements and stories were assigned low feature values by key stakeholders and, consequently, were pushed off to later releases (some to the end of development). The result of this, as described by interviewees, was that the project teams pushed the security requirements release increments very late in the life cycle and then tried to add in a security infrastructure after development was almost complete. In one interview, this scenario was referred to as "bolting on security." Adding the security features late usually results in devastating changes to the architecture causing schedule delays and cost overruns. These situations often occur, said one interviewee, because backlog is prioritized based on value alone and not value and risk. Like the project managers we interviewed, information assurance interviewees said that accrediting authorities are key stakeholders who need to be involved early in the development life cycle to help prioritize requirements.

Agile projects may not have security design documentation if they follow evolutionary architecture practices.

A problem articulated by some of the information assurance interviewees is that they need enough architecture documentation to reason about delivering security risk (particularly if the project team wants to deliver security features incrementally). In some cases, however, Agile project teams follow an evolutionary architecture process so they not do as much upfront architecture design and documentation as traditional development projects. Consequently, the security design produced in an evolutionary manner may not be detailed enough for accreditation reviewers to evaluate the design for vulnerabilities. For this reason, additional upfront design work may be necessary for Agile projects. We suggest that the detailed security design should be planned and implemented in Phase 0 as part of the architectural runway [Leffingwell 2011]. In addition, the security design must also be consistently maintained. A current set of architectural views is help-

ful for reasoning about the impact of adding a feature on the security posture (negative or positive). It also helps with assessing ability of the design against new emerging threats as well as defect root cause analysis. Root cause analysis may also require Agile teams to think about backlog from a more holistic viewpoint, as described in the quote below.

> *"We had to group backlog intelligently so that we could reason about root causes and come at problems with a holistic solution. This is a little different from the normal Agile approach where you just pick features off the backlog."*
> *—industry Agile project lead*

Challenges "finding the architect" (someone accountable for the security design).
One of the fairly well-accepted Agile best practices is the notion that small, dynamic teams are preferred over large, static teams. If the project is relatively small (two or three teams), it may be possible to successfully share the responsibility for designing and maintaining the security design as a team of peers. However, the shared approach to managing the security design becomes more challenging on large projects. With shared architectural responsibility, it can become difficult to determine who is accountable for the overarching security design. In addition, there may be a lack of clarity regarding the impact of changes that are cross-cutting between teams. To address this issue, one interviewee said that, while it is not commonly accepted practice on all Agile projects, they have a single overarching software chief architect who is responsible for the overall high-level software design—including the security design. Others we interviewed have had success with a shared architecture team approach as long as they had well-defined roles and responsibilities. Regardless of how this problem is addressed, it is necessary for Agile project teams to explicitly assign the responsibility of managing and maintaining the security design to someone.

Dealing with the possibility of incremental release feature slip.
Some interviewees said that an inhibitor to the idea of delivering security features incrementally is the possibility that the scheduled features may not be delivered in the planned increment. Agile project teams often maintain velocity for a while, but for reasons that are often beyond the control of the project, velocity may slow and features may slip to the next increment. The problem is that if the project team makes an agreement with the accrediting authorities that they will implement a set of security features within a certain timeframe, a "slip" can damage the team's credibility and increase security risk. It is important that Agile project teams stick to their commitments or actively involve the information assurance representatives in re-planning. The information assurance stakeholder needs to remain involved in subsequent increment planning cycles to ensure the security requirements remain a high priority.

Accrediting authority independence and an Agile collaborative environment.
Earlier in this paper we discussed the suggestion several interviewees made to have someone with accreditation expertise embedded on the Agile project team—especially early in the design phase. Like the project managers we interviewed, interviewees representing the accreditation organization perspective also supported the notion of having the accrediting authorities support two roles; a consulting role and a separate independent reviewer role. The interviewees representing the accrediting authority particularly emphasized the importance of having this support early in the de-

sign and development life cycle. The information assurance interviewees emphasized the need to maintain accreditation reviewer independence and objectivity, preferably by avoiding having the security consultant and reviewer be the same person.

8 Implications of Existing Information Assurance Processes on Agile Principles

With a quick glance at the list of 12 Agile principles [Agile 2011], it becomes quickly evident that several may be potentially hindered by the DIACAP life cycle shown in Figure 1. Some of these principles are listed below followed by a short impact statement. These impact statements have been corroborated using data from the interview findings described in the subsequent section.

- *"Our highest priority is to satisfy the customer through early and continuous delivery of valuable software."* DIACAP supports security requirement prioritization to some extent with the CAT 1, 2 and 3 vulnerability rating structure. However, because security is treated as a separate entity with its own processes and policies, it is difficult to discuss tradeoffs and prioritize security requirements with other backlog. Security requirements continue to be treated as separate from other requirements. Perhaps better integrating the security process with other feature development is an area for future work.
- *"Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage."* Only a major change in the information assurance posture should require a reaccreditation. However, DIACAP is not often interpreted this way. When changes are made to a system, often a full reaccreditation is required. This can take months or years. This can significantly hinder the Agile project's ability to rapidly adapt to changing requirements. One interviewee suggested that the DIACAP controls talk about change, and documenting how you address change, but the problem is many systems fall short on this control. In fact 8510.01 supports the notion that the security process should allow for change, stating: "The approval to operate accreditation decision should not be reserved for DoD information systems for which no change is planned or foreseen. Such thinking engenders an abuse of the IATO [interim authorization to operate] accreditation status and is an inaccurate portrayal of the DoD information systems information assurance posture."
- *"Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale."* Again, delivering working software within a few weeks is challenging with a waterfall-based accreditation process such as DIACAP. This problem is compounded by that fact that the DIACAP accreditation process is largely a manually intensive review process.
- *"Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done."* The good news is that the DoD IA community is starting to embrace the notion of providing tools and resources to help project teams integrate security into the development process. For example, organizations such as DISA are releasing helpful tools that project teams can leverage to incorporate continuous vulnerability scans into their regression testing processes. The bad news is that the "trust them to get the job done" part is still a challenge area. The current DIACAP is still very much a review step that is imposed by an external body. One interviewee pointed out the following: "Trust must be earned. If projects took security more seriously DIACAP would never have been born. Projects have

proven they cannot be trusted in addressing security without some forcing function. It is not realistic, or prudent, to suggest that DoD programs adopt an entirely self-governing posture toward security accreditation." Perhaps we can move to more of a *trust but verify* model over time by embedding an information assurance stakeholder as a contributing part of the team for a period of time, as suggested by several interviewees.

- *"Continuous attention to technical excellence and good design enhances agility."* The fact that the DIACAP accreditation step is executed as a gating function at the end of the life cycle can promote a "we'll deal with security at the end" mentality toward information assurance. This is a big problem for a couple of reasons. First, as with any quality attribute, security architecture can't be "bolted on" at the end [Boehm 1988]. While in an Agile context the architecture may evolve with each increment as opposed to being fully fleshed out at project start, there still needs to be foundational security architecture for future increments to build on. Second, the check-the-box mentality may encourage engineers to apply controls in a wholesale fashion rather than reasoning about the appropriateness of a particular control in their particular design. In other words, if we are not careful the DIACAP may actually discourage good engineering and design practices rather than encourage them.
- *"Simplicity—the art of maximizing the amount of work not done—is essential."* In order for Agile projects to succeed, it is important to constantly be on the lookout for opportunities to avoid doing unnecessary work. DoD Agile project teams need to assess every requirement, including security requirements, for appropriateness and work with accrediting authorities to investigate options for dealing with controls that may not be relevant to their circumstance.

9 Looking Ahead: What to Expect Five to 10 Years from Now

The use of Agile methods has been successfully integrated with C&A in selected DoD programs to improve the rate of delivery of technology capabilities to the operational environment. What are the possibilities for the five to 10 years ahead?

9.1 DoD Concern for Operational Security Grows

DoD concern for operational security is growing and there is no reason to suspect the concern will taper off. The DoD push to outsource, relying more heavily on off-the-shelf products (commercial off the shelf, government off the shelf, open source) and vendor supply chains, while attractive in terms of acquisition cost and schedule, brings with it security vulnerabilities which have been prevalent in the commercial market place for over a decade. The DoD has been able to avoid many of these security problems in the past by building its own solutions that meet the unique needs of its threat environment, but this option is coming to a close.

In the commercial environment, "the evolution of computer abuse—and therefore of computer security—is driven by commerce. Botnets, spam, phishing, banking Trojans, identity theft and so on are all commercially motivated enterprises perfected in a constant arms race with a well-financed computer security industry [Savage 2011]." In joining this environment, DoD will be supplying an extremely large and attractive target for criminal and nation-state adversaries.

This trend is further enhanced by the DoD desire to expand into the use of mobile and cloud capability—highly attractive commercial offerings with increased flexibility and functionality that support an increasingly technology-savvy user population. Because these technologies are developed for the commercial world, these capabilities have not been built to address the unique protection needs of the DoD. Commercial organizations are struggling to determine effective operational security responses and take advantage of the flexible functionality. Problems with these technologies such as the Amazon cloud failure in 2011 [Amazon 2011] and Blackberry outages in October 2011 [RIM 2011] and again in March 2012 [RIM 2012] raise questions of reliability and stability which can link to security problems.

9.2 DoD Information Assurance Controls Strengthen

DoD operational support is being asked to provide an operational environment that continues to function at low risk when the systems and software being pushed into that environment carry increased security challenges. The current evaluation process, occurring just prior to deployment, does not provide sufficient insight for timely and cost effective adjustment if there are problems. The Program Protection Plan, a revised policy for certification and accreditation released July 18, 2011 by the Principal Deputy Under Secretary of Defense for Acquisition, Technology, and Logistics (PDUSD(AT&L)), identifies steps of the evaluation that should occur at each acquisition milestone to provide earlier intervention opportunities into the acquisition process [DAU 2011].

Congress is focused on mandating DoD improvements to software assurance which are linked to the DoD National Defense Authorization Act of Fiscal Year 2011. Public Law 111-383, dated

Jan. 2, 2011, per section 931 [NDAA 2011] “Requires the Secretary to direct DoD's Chief Information Officer to work to achieve: (1) the continuous prioritization of the policies, principles, standards, and guidelines developed under the National Institute of Standards and Technology [NIST] Act with agencies and offices operating or exercising control of national security systems based upon the evolving threat of information security incidents with respect to national security systems, the vulnerability of such systems to such incidents, and the consequences of such incidents; and (2) the automation of continuous monitoring of the effectiveness of the information security policies, procedures, and practices within the information infrastructure of DoD, and the compliance of that infrastructure with such policies, procedures, and practices.” To improve alignment with NIST, the DoD is planning to update DIACAP to align with the NIST risk management framework.[INFOSEC 2011]

This same legislation, per section 932, requires the DoD to develop and implement “a strategy for assuring the security of software and software-based applications for all covered systems.” This strategy is required to include considerations for continuous monitoring of operational software and validations earlier in the life cycle. The strategy must also include “Mechanisms for protection against compromise of information systems through the supply chain or cyber-attack by acquiring and improving automated tools for—(A) assuring the security of software and software applications during software development; (B) detecting vulnerabilities during testing of software; and (C) detecting intrusions during real-time monitoring of software applications” [GPO 2011].

Extensive consolidation of DoD networks including management and monitoring is underway to improve the flow of critical data within and among the military services. The importance of the DAA in controlling the flow of operational risk into these highly sensitive operational environments is growing. The Army has recently moved from 200 to 52 DAAs, vastly increasing their span of control, and similar consolidations have been reported by the other services. The Navy also consolidated DAA authority to a single flag-level DAA. These individuals will benefit from programs prepared to support their decision-making with effective and transparent risk management approaches that provide evidence and justification to support an authority to operate request.

9.3 Agile Approaches are Well Positioned to Address the Future

The increased visibility of operational security and the growing legislative focus on software assurance for DoD are driving change into the certification and accreditation process. This represents an opportunity for Agile projects to demonstrate the ways this development approach can provide added value in improved software assurance in addition to faster feature delivery for the customer. However, it will require some thoughtful considerations of the needs of the DAA as a unique stakeholder that build confidence that the outputs have effectively addressed operational security. C&A cannot be treated as a separate parallel activity; plan for C&A the same way that one plans for stakeholder delivery, but the negotiations will be with a different group of stakeholders (DAA and C&A) and must be based on a different set of values—operational risk.

A program applying an Agile approach will be focused on establishing ways to continuously test the system from the start. Confidence in the secure quality of the code can be enhanced by expanding the testing to include tools such as compilers that identify security coding problems, static analysis tools to identify vulnerabilities, and fuzz testing tools to extensively explore ways in

which bad data can stress the code outside of normal ranges—validating that it does not respond inappropriately. As DAAs see that this level of validation occurs as part of the standard way of doing business, their confidence in the security quality of the code presented to them for approval will increase.

Regression testing is performed continuously across code releases. As a result, problems identified and fixed in early releases will not reappear in later ones. The collection of data that shows the progress made by the Agile team in addressing security vulnerabilities as they build code through multiple releases provides evidence that will increase the DAAs' confidence.

Agile provides mechanisms to address operational problems quickly in subsequent releases. Establishing ways for DoD operational problems to be added to the backlog, as well as tracking the pace at which corrections address operational needs are implemented, will support a perspective that C&A does not need to exhaustively test to prevent every possible operational security issue. This way incremental security improvement can occur in tandem with expanded feature development. This is not feasible with other approaches and developers are typically gone by the time security breaks after implementation. Support for future maintenance is often not planned for—so pushing off the security requirements can generate mounting technical debt for later maintainers.

Security expertise needs to become part of the Agile project resources. It may not be feasible or practical to have a security expert working full time on the team, but the team needs to have access to this knowledge to be effective. At a minimum, all developers need a basic level of understanding as to why security is important and of the potentially devastating operational problems that security vulnerabilities create for the DoD. To achieve this, it is imperative developers of DoD software read the DISA Application Security and Development *STIG* as a bare minimum to understand what a Category 1 vulnerability is and why no software should be released with these vulnerabilities as well as understanding what ports, protocols, and services are prohibited by DoD. In addition, developers should understand how their code can contribute to security problems and what they should be doing to avoid these problems. Monitoring code testing to track the identification and correction of security vulnerabilities provides evidence to the DAA that the project is serious about addressing C&A needs.

Further security capability should be provided by someone with an in-depth knowledge of the DIACAP who would monitor team outputs to identify security problems early and track effective mitigations. Code to address security controls could be built over several releases, but the version that is offered for operational implementation needs to fully address operational security.

Ways in which dashboard reporting could be used to exhibit security problems identified and addressed could provide further evidence to the DAA. Trends in security problem data could be useful in demonstrating when a product is ready for C&A based on a reduction in the security problems identified. Recent research [Brune 2012] has shown the value in tracking change and error rates to predict when products are ready for release and a stronger use of metrics in tracking vulnerabilities could be applied in information assurance risks.

To recap, in this section we articulated these future concerns in the area of security followed by ways in which these concerns could be addressed. We assert that with thoughtful integration of C&A needs Agile could become the development approach of choice by security decision makers.

10 Closing Thoughts

This paper addresses the challenges Agile project teams face as they try to deliver capability rapidly while at the same time trying to adequately meet the critical requirements of the DoD information assurance accreditation process. In summary, this paper presents results from a very short literature search, as well as information gathered through a series of interviews on this topic. The interviews were conducted with the DoD Agile program managers (and teams) and active members of the DoD information assurance accreditation community. The information provided in the interview sections focuses on what Agile program managers and teams need to do to handle the DoD certification and accreditation process today. The interview data was synthesized to produce this set of recommendations:

- Define criteria for reaccreditation early in the project.
- Leverage long accreditation approval wait time with frequent community previews.
- Ensure requirements are prioritized according to value and risk.
- Make sure Agile project teams understand the intent behind security requirements.
- Ensure Agile development processes produce "just enough" security architecture.
- Make sure there is at least one person with strong security analysis expertise on the team.
- Foster Agile project team and accrediting authority collaboration.
- Don't apply all the information assurance controls blindly.
- Use COEs, SDKs and enterprise services to reduce accreditation time.
- Apply a risk-based, incremental approach to security feature design and development.
- Leverage architecture tactics such as layering and encapsulation to minimize impact of change.
- Leverage unclassified environments for development and community previews.
- Encourage the Agile teams to merge Agile and security best practices.

In addition, we included a section looking five to 10 years ahead into the DoD cyber security environment of the future. As we "look ahead" we discuss expectations for growing future cyber threats, as well as suggestions for dealing with these coming challenges.

There is much additional work that could be done in this topic area. Here are some suggestions for future work:

- further technical research in some of the key challenge areas listed in this report
- in-depth analysis of information assurance future policy
- additional scope to cover implications of Agile on testing

Our hope is that by providing these lessons learned and helpful suggestions based on real-world experience program managers can better prepare for and address these challenges.

References/Bibliography

URLs are valid as of the publication date of this document.

[AFEI 2012]
The Association for Enterprise Information. Section 804 Task Force.
http://www.afei.org/WorkingGroups/section804tf/Pages/default.aspx

[Agile 2011]
Principles behind the Agile Manifesto
Retrieved May 24, 2011, from Manifesto for Agile Software Development:
http://agilemanifesto.org/principles.html

[Amazon 2011]
Summary of the Amazon EC2 and Amazon RDS Service Disruption in the US East Region.
http://aws.amazon.com/message/65648/

[Army 2010]
Common Operating Environment. Army CIOG6 website.
http://ciog6.army.mil/ArmyEnterpriseNetworkVision/tabid/79/Default.aspx

[Bass 2003]
Bass, Len, Clements, Paul, & Kazman, Rick. *Software Architecture in Practice, 2nd Edition*. Addison-Wesley, 2003.

[Bellomo 2011]
Bellomo, S. *A Closer Look at 804: A Summary of Considerations for DoD Program Managers* (CMU/SEI-2011-SR015). Software Engineering Institute, Carnegie Mellon University, 2011.
http://www.sei.cmu.edu/library/abstracts/reports/11sr015.cfm

[Boehm 1988]
Boehm, Barry. "A Spiral Model of Software Development and Enhancement." *Computer 21*, 5 (May 1988): 61-72.

[Brown 2010]
Brown N., Nord R, & Ozkaya I. "Enabling Agility Through Architecture," *Crosstalk* (Nov/Dec 2010): 12-17.

[Brune 2012]
Brune, Philip & Wild, Roman. "Determining Software Product Release Readiness by the Change-Error Correlation Function: On the Importance of the Change-Error Time Lag," 5360-5367. 45th Hawaii International Conference on System Science, Maui, HI, January 2012.

[CERT 2012]
Secure Coding. CERT website.
http://www.cert.org/secure-coding/

[Clements 2011]
Clements, Paul C., Bachman, Felix, Bass, Len, Garlan, David, Ivers, James, Little, Reed, Merson, Paulo, Nord, Robert, & Stafford, Judith A. *Documenting Software Architectures: Views and Beyond, Second Edition*. Addison-Wesley Professional, 2011.

[Coverpages 2012]
Application Security Standards. Coverpages website.
http://xml.coverpages.org/appSecurity.html

[DAU 2011]
Program Protection Plan. Defense Acquisition University website.
https://acc.dau.mil/CommunityBrowser.aspx?id=322418

[DIACAP 2012]
Welcome to the DIACAP Implementation Portal. DIACAP Implementation website.
http://www.DIACAP.org

[DISA 2011]
DISA. *Application Security Technical Implementation Guide*. Version 3, Release 4; 28 October 2011.

[DoDI 2008]
DISA. *DoDI 8500-2 IA Control Checklist - MAC 2-Classified*. Version 1, Release 1.4; 28 March 2008.

[DoD 2009]
Department of Defense. *Report of the Defense Science Board Task Force on Department of Defense Policies and Procedures for the Acquisition of Information Technology*. Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics. March 2009.

[DoD 2010]
Department of Defense. *Interim Acquisition Guidance for Defense Business Systems (DBS).* Memorandum released November 15, 2010.

[DoD 2010a]
Department of Defense. Press Release: *Report on New IT Acquisition Process Released*. Dec 9, 2010.
http://dcmo.defense.gov/documents/Report_New_IT_Acquisition_Process.pdf

[GPO 2011]
Government Printing Office. *Public Law 111–383*. Jan. 7, 2011
http://www.gpo.gov/fdsys/pkg/PLAW-111publ383/pdf/PLAW-111publ383.pdf

[INFOSEC 2011]
Goodbye DIACAP, Hello DIARMF. INFOSEC website.
http://resources.infosecinstitute.com/goodbye-diacap-hello-diarmf/

[Lapham 2010]
Lapham, Mary Ann, Williams, Ray, Hammons, Charles, Burton, Daniel, & Schenker, Alfred. *Considerations for Using Agile in DoD Acquisition* (CMU/SEI-2010-TN-002). Software Engineering Institute, Carnegie Mellon University. April 2010.
http://www.sei.cmu.edu/library/abstracts/reports/10tn002.cfm

[Leffingwell 2011]
Leffingwell, D. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Pearson Education Inc., 2011.

[NDAA 2009]
National Defense Authorization Act for Fiscal Year 2010, 10 U.S.C., Pub. L. 111-84 § 804. 2009.

[NDAA 2010]
National Defense Authorization Act of 2011, P. L. 111-383, Section 804SEC. 804.

[NDAA 2011]
National Defense Authorization Act of Fiscal Year 2011. Public Law 111-383, section 931.
http://www.govtrack.us/congress/bills/111/s3454

[NDIA 2011]
National Defense Industrial Association. *Agile (Scrum) Workshop*.
http://www.dtic.mil/ndia/2011agile/Agenda.pdf

[RIM 2011]
Research in Motion. *BlackBerry Issues Statement Over Downed Services*. Oct. 11, 2011.
http://www.zdnet.com/blog/btl/blackberry-issues-statement-over-downed-services/60450?tag=content;siu-container

[RIM 2012]
Research in Motion. *RIM Lost $54 Million on Four-Day Global BlackBerry Outage*. March 30, 2012.
http://www.zdnet.com/blog/london/rim-lost-54-million-on-four-day-global-blackberry-outage/3736

[Savage 2011]
Savage, Stefan. “In Planning Digital Defenses, the Biggest Obstacle Is Human Ingenuity.” *New York Times* website, December 5, 2011.
http://www.nytimes.com/2011/12/06/science/stefan-savage-girding-for-digital-threats-we-havent-imagined-yet.html

[Section 804 2012]
A New Approach for Delivering Information Capabilities in the Department of Defense. The Section 804 Task Force, The Association of Enterprise Information website. http://www.afei.org/WorkingGroups/section804tf/Pages/default.aspx

[Security 2005]
Seclists.org website, blog posting Sat, 26 Nov. 2005 http://seclists.org/pen-test/2005/Nov/237

[US-CERT 2012]
Build Security In. US-CERT website. https://buildsecurityin.us-cert.gov/daisy/bsi/home.html

[West 2011]
West, Dave. *Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today*. Forrester, July 26, 2011.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	November 2012	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
DoD Information Assurance and Agile: Challenges and Recommendations Gathered Through Interviews with Agile Program Managers and DoD Accreditation Reviewers	FA8721-05-C-0003

6. AUTHOR(S)

Stephany Bellomo, Carol Woody

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2012-TN-024

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFLCMC/PZE/Hanscom Enterprise Acquisition Division 20 Schilling Circle Building 1305 Hanscom AFB, MA 01731-2116	

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

This paper was produced by the Software Engineering Institute at Carnegie Mellon University in support of the Agile acquisition research agenda funded by the Office of the Secretary of Defense. This paper is part of a larger research study focused on understanding the implications of applying a rapid, incremental development approach, such as Agile, on the Department of Defense (DoD) acquisition process. An overarching goal of this research agenda is to identify areas of tension between Agile and existing processes and provide recommendations for improvement to those processes. In support of the overarching research agenda, several "point" papers are being developed on particular topic areas. The topic of this particular paper is the natural tension between rapid fielding and response to change (characterized as agility) and DoD information assurance policy. The authors gathered information for the paper primarily by conducting interviews with several DoD project managers and information assurance representatives. The interview findings are organized into a list of key challenges and recommendations. The paper also includes a five- to ten-year future outlook with respect to information assurance and agility in DoD. The opinions, findings, conclusions, and recommendations expressed in this Technical Note are those of the authors and do not necessarily reflect the views of the United States Department of Defense.

14. SUBJECT TERMS	15. NUMBER OF PAGES
acquisition, Agile, information assurance	46

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18
298-102