8-2015

Agile Information Security Using Scrum

James R. Fitzer

APUS Library Capstone Submission Form

This capstone has been approved for submission to and review and publication by the APUS Library.

Student Name [Last, First, MI] *	Fitzer	James	
Course Number [e.g. INTL699] *	ITCC698	**Paper Date** [See Title pg.]	05/2015
Professor Name [Last, First] *	Dr. Watson-Stone, Novadean		
Program Name * See list	Master of Science in Information Technology		
Keywords [250 character max.]	Agile development, information security, information protection, Sc		
Passed with Distinction * Y or N	Y		
Security Sensitive Information * Y or N	N		
IRB Review Required * Y or N	N	If YES, include IRB documents in submission attachments.	
Turnitin Check * Y or N	Y	All capstone papers must be checked via Turnitin.	

* Required

Capstone Approval Document

The thesis/capstone for the master's degree submitted by the student listed (above) under this title *

AGILE INFORMATION SECURITY USING SCRUM

has been read by the undersigned. It is hereby recommended for acceptance by the faculty with credit to the amount of 3 semester hours.

Program Representatives	Signatures	Date (mm/dd/yyyy)
Signed, 1st Reader * [capstone professor]		
Signed, 2nd Reader (if required by program)		
Recommendation accepted on behalf of the program director *	Novadean Watson-Stone Digitally signed by Novadean Watson-Stone DN: cn=Novadean Watson-Stone, o=American Public University System, ou=APUS, email=nwatson@apus.edu, c=US Date: 2015.07.07 11:18:23 -04'00'	
Approved by academic dean *	Dan Benjamin Digitally signed by Dan Benjamin Date: 2015.07.08 11:56:49 -04'00'	

* Required

AGILE INFORMATION SECURITY USING SCRUM

A Master Creative Project

Submitted to the Faculty

of

American Public University

by

James Richard Fitzer

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

August 2015

American Public University

Charles Town, WV

DEDICATION

This work is dedicated first and foremost to my wife and daughter, whose support and love have kept me sane. And to the developers, management, and my team of engineers at the Northern Border Integration Demonstration, whose advice and hard work made this book possible.

ACKNOWLEDGMENTS

I would especially like to thank Dr. Novadean Watson-Stone, whose guiding hand throughout the process made this project possible. I would also like to thank the entire faculty during my long time at AMU; they are consummate professionals whose guidance not only led to my success, but developed the very ideas and questions that lead to this project.

I would finally like to thank Robert Dibble, whose leadership during NBID, and continued guidance throughout this project, helped define just what an Agile information security program should be.

ABSTRACT OF THE PROJECT

AGILE INFORMATION SECURITY USING SCRUM

by

James Richard Fitzer

American Military University, August 2015

Charles Town, West Virginia

Dr. Novadean Watson-Stone, Thesis Professor

The increased importance of information protection, coupled with rapidly changing security landscape, has led to information security professionals finding it difficult to stay ahead of emerging threats. This problem has been compounded by the increased prevalence of agile software development methodologies, which ensure a rapidly changing system. This project unifies the principles of Agile software development, particularly Scrum, with established security best practices in the form of a technical book for mass-market publication. The book provides a guidance and framework for using the Scrum method to construct an information security program, conduct risk assessments, and implement policies and controls. This is accomplished through a review of existing knowledge on the topics of agile development and information security, and the author's work to unify the two. Additionally, it provides guidance on leading such a team, and lessons and anecdotes from the author's experience conducting security management on the Northern Border Integration Demonstration (NBID), an agile project. The book will be published through mainstream retail outlets and presented at conferences and events.

Keywords: Agile development, information security, information protection, Scrum

TABLE OF CONTENTS

LIST OF TABLES

TABLE PAGE

Introduction

One of the chief problems within the Information Security discipline is the need to balance the business or product with information security and assurance goals. All too often (especially in government or financial sectors) information security personnel, developers, and system administrators and engineers are at odds with one another. The entire academic program within the Information Security concentration here at American Military University has focused on bridging those gaps, ensuring that Information Security professionals are constantly reminded of not only their importance, but their place. Indeed, the very first chapter in the first information security text of the Information Assurance concentration states:

> Information Protection should support the business objectives or mission of the enterprise. This cannot be stressed enough. All too often, information security personnel lose track of their goals and responsibilities. The position of ISSO (Information Systems Security Officer) has been created to support the enterprise, not the other way around. (Peltier, Peltier, & Blackley, 2005, p. 1)

This statement, so early in the program, is repeated time and again, and should serve as a guiding principle for Information Security professionals, where operational requirements and security guidelines can collide.

The popularity of various Agile systems of software development has greatly accelerated over the past few years, with VersionOne reporting that 49 percent of businesses say that most of their company is using Agile development, with 52 percent reporting Scrum as their most popular Agile method ("7th Annual State of Agile Development Survey," 2012).

The Problems with Agile Software and Security

The difficulty comes when uniting Agile software development with Information Security. Bejtlich describes Security as the process of maintaining an acceptable level of

perceived risk, and Dr. Mitch Kabay of the International Computer Security Association says that “security is a process, not an end state” (Bejtlich, 2005, p. 4). However, recent high level breaches, advancement of regulations for the protection of data (particularly Personally Identifiable Information, or PII), and the sheer explosion of internet-connected devices and systems have brought an enormous amount of attention to the field of Information Security, resulting in not only greatly increased focus on countermeasures, but a “bloat” in bureaucracy and process. This is nowhere more apparent than within the Department of Defense, where Information Assurance (IA) often ignores that very basic first tenet of supporting the mission rather than becoming it.

Security is seen as “a bottomless pit of costs—at best, a necessary evil,” and “is generally perceived as a constraint to business” (Brotby, 2009, p. xv). During the Northern Border Integration Demonstration (NBID), the NBID team observed first-hand the clash between security and mission focus; between endless documentation and agile software. The problem unifying agile software development with traditional information security objectives is daunting, but not insurmountable, and over the years the NBID team spent on that project, workable solutions were found by distancing the team from the traditional mindset, and working agility into the NBID security process.

The problem can be summarized as follows: Although there is much literature on Agile Software Development using Scrum, and Information Security, there exists a gap in the literature attempting to unify the two. I intend to fill that gap with a practical, useful handbook that provides a workable solution to the seemingly opposing goals of agility and security.

Project Goals and Relevance to the Discipline

When conducting a security risk assessment for a review of system accreditation with NBID, Landoll’s “The Security Risk Assessment Handbook” was used extensively. The

processes and artifacts described in the text represent a very complete and exhaustive guide to understanding the threats in any environment, determining relevant controls, and implementation of these controls to minimize residual risk. Unfortunately, these artifacts took the NBID security team so much time to produce, that by the time they were delivered, the software build in question was already deployed and the next had begun. The nature of Agile development with a framework like Scrum requires a time-boxed "sprint," whose end date does not change. The build is delivered on that date, and any portions that are not complete get removed and put back into the backlog. The pace of such a project means that IT security personnel are forever catching up to the latest release.

It is my intent to produce a book that can serve as a useful guide for IT security professionals who need to maintain accreditations, provide risk assessments, and implement controls in an Agile development environment. Using this research as a framework for required material, this book will be a reference that unifies Agile software development with compliance-based IT security needs, in a format that is not only accessible to a security professional, but becomes a reference that is as frequently referred to as Landoll's and Bejtlich's works were to me. The text shall be released online not only through the APUS library, but through as many other mediums as possible (including print via Amazon), furthering the goal of providing a timely, useful reference to as many as possible within our discipline. In the interests of furthering the Agile mindset, the product will be distributed for low cost, to further the understanding of the specific and unique requirements of securing software in an Agile environment.

Sponsor

The sponsor for this project will be Cory Burns, who is the Live Site Infrastructure Engineering Lead for the SharePoint Online service at Microsoft. He is assisting with the tracking of project deliverables, ensuring my deadlines as per my project schedule are met, and

proofreading/providing feedback on the text as it comes. Cory represents one of the target audiences for the text: an engineer in charge of infrastructure and services in a rapidly changing environment. His position outside of the security field will assist in determining readability, as the text should be accessible to those who are not subject matter experts in security.

Table 1 - Project Schedule and Deliverables

		Timeframe	
		Start	End (due)
Book Chapters	Primer on Agile Development		March 23
	Need for Agility in Security		March 24
	Case Study Introduction: Northern Border Integration Demonstration (NBID)		March 25
	Obstacles to Agile		March 26
	Scrum Introduction		March 27
	Coordinating Developers and Security Staff		March 28
	NBID: Balancing Rapidity and Compliance		March 29
	Scrum Roles in Information Security		March 30
	Sprints		March 31
	Daily Meetings / Scrum of Scrums		April 1
	NBID: Scrum Standup and IT Staff		April 2
	Artifacts		April 3
	Documentation in Agile		April 4
	Security Risk Assessment as a Living Document		April 5
	Regulation Compliance vs. Agility		April 6
	NBID: When Security and Operational Requirements Collide		April 7
	Prioritization / Vulnerability Management		April 8
	Prevention vs Monitoring in an Agile Environment		April 9
	Preproduction Systems Security		April 10
	Resource Management Concerns		April 11
	Conclusion		April 12
Publication Prep	Cover Art, Chapter, and Paragraph Formatting	April 20	April 22
	Footnotes	April 20	April 23
	Draft Project Submission		April 26
	Final Approval from Bob Dibble (Navy) for NBID content	April 20	April 26
	Correction via feedback from Professor	April 27	May 10
	Final Project Submission		May 17

Project Plan Acceptance Criteria

The project must meet the following criteria in order to meet the goals of the project:

- Provide the reader an acceptable background on Scrum and Agile development.
- Describe the NBID project in enough detail to be relevant, but meet with Naval Surface Warfare Center representative Bob Dibble's approval to ensure no classified or sensitive data is released.
- Adequately provide the reader sound guidelines for conducting information security activities in an agile development project, rapid integration project, or similarly fast-changing environment.
- Be ready to publish in traditional textbook or industry handbook format.

Final Deliverable

The final deliverable will be the compilation of all chapters listed in the schedule into a formatted, publication ready handbook industry standard formatting and style, submitted in Word format, and converted to PDF for publication as an eBook, and published via Amazon CreateSpace for print.

Preliminary Research

In the spirit of bringing the entirety of my coursework to its conclusion, the previous research conducted into various information security topics is extensive. In addition to the works that have become indispensable such as Landoll's handbook and the work of Bejtlich, a great deal of time has been spent researching implementations of Agile methodologies. Scrum has been chosen not only because of its commonality within the Agile development world, but also previous experience with it at NBID, and a wide variety of reference material on the methodology.

Further Questions

- Can an Agile Information Security framework be constructed with the following parameters?
    - Flexible enough to be modified to various situations/projects
    - Rigid enough to provide solid and useful guidance; functioning as a “how to” guide on implementing security with Agile methods/

The sheer magnitude of variations within such a program raise the challenge of providing enough detail to be comprehensive, without becoming a cumbersome tome that not only exceeds the scope of this program, but exceeds a reasonable length to be considered a “quick reference” by industry professionals. The conformity with discipline standards is ensured by studying the literature not only for content, but format and convention: producing an IT handbook absolutely must have a solid foundation in the literature and research, but must be flexible and adaptable to be of use. The research method for the text will be primarily a review of existing literature to identify pertinent best practices in agile software development (including timelines, meetings, and team composition), and use those practices to satisfy the goals of a typical IT security program.

Goals and End State

Through exhaustive examination of available literature on Agile, the desired end state of this project is a publication-ready text, made widely available, that provides IT security professionals with a workable framework for Agile security management, including not only the crucial development of risk assessments, but also the implementation of controls. This will be done not only through established Agile wisdom and IT security fundamentals, but through the experiences and challenges gained by working and securing an Agile government IT project.

Literature Review

Introduction

The nature of creating a text for information security professionals that is operationally useful requires not only a thorough understanding of the topical areas, but a demonstrable need for the project. The utility of principles developed by the author through personal experience do not necessarily reflect an emerging or existing need within the information security field, and thus a review of the available body of knowledge within the intersecting disciplines of agile development and information security is necessary.

The prime goal of this review is to establish the utility of the project by examining the state of Agile development in the enterprise, as well as the presence of existing Agile-related literature that is indicative of current use of the philosophy; however, ancillary goals include a review of Agile development practices to ensure the final text gives proper and meaningful background, as well as an examination of security literature to ensure the methods formulated by the project adhere to at least the spirit, if not the letter, of conventional information security practices. Finally, the review will examine other efforts to extend Agile principles onto other activities, and provide context directly related to specific examples (both NBID and the government as a whole) that will be covered in the final text. The reader of the text produced will assume expertise on the part of the author, which speaks to the necessity of a review of Agile and information security literature, as well as the examination of the relations between those topics. Such content will not be included in the final published book; however this examination provides the lynchpin by which the content of the book is constructed.

Development of rapid systems requires a fast-moving team, and change is the only constant. By examining Agile development practices, and applying those principles to established information security principles, a picture of Agile information security emerges. The

goal of such a philosophy is to provide resources to information security professionals to help them avoid constantly falling behind in their security posture, and to emphasize response and mission focus over the constant fight against emerging exploits on a reactionary basis. To that end, the first examination must be toward the prevalence of Agile development within the enterprise.

Use of Agile Development in the Enterprise

Results. VersionOne, a company that produces project management software oriented towards Agile projects and teams, has conducted an annual "State of Agile Development" survey each year since 2006. In it, respondents are asked specific questions related to the development activities within their organization, their own knowledge and background with Agile methods, and institutional and organizational obstacles and challenges to Agile development. Through this survey (which generally raises comparisons with previous years' data), a broad overview of Agile development can be gleaned, although it is not without bias, which is addressed later.

In order to establish the project's need, the state of Agile development in the enterprise, and its prevalence, must be examined. More than half of respondents are managing between 50 and 100% of their projects using Agile ("8th Annual State of Agile Development Survey," 2013, p. 2). Nearly 40% of respondents are managing at least 75% of their projects using Agile. In total, 88% of respondents said their organizations were using Agile development in the enterprise, which is an increase from 84% in 2012 and 80% in 2011 ("8th Annual State of Agile Development Survey," 2013, p. 2). This data suggests that Agile development continues to be a strong force within the enterprise.

Although the prevalence of Agile development is fairly well established by several years of these surveys, particularly interesting are the questions regarding the initiation of Agile methods and their origination. According to the survey, the primary champions of Agile methods

(those making decisions about the use of Agile) are management personnel with 61% ("8th Annual State of Agile Development Survey," 2013, p. 3). Additionally, executives account for 14%, and "Consultants/Trainers/Other" account for 8%, leaving a mere 17% classified as developer and IT staff. Survey demographic data reveals that the respondents consisted of 48% project management, Scrum Master, and team lead roles, with 26% combined between development and IT staff ("8th Annual State of Agile Development Survey," 2013, p. 1). Further, Leadership is rated by the survey as the personnel "most knowledgeable about Agile," whereas IT staff are not even registering a single percentage ("8th Annual State of Agile Development Survey," 2013, p. 2). The presence of IT staff within demographic data and initiation data, and the lack of such staff from the expertise results at all, suggests a gap of knowledge between management, developers, and IT staff regarding Agile development; a problem that is one of many this project seeks to correct.

The choice of Scrum as the primary focus for the product seems justified: Scrum and its variants account for 73% of the Agile methodologies used by respondents. Additionally, a hallmark of Scrum, the "daily standup," a meeting of current status so named because it is performed standing up (to ensure brevity) is used by 85% of respondents ("8th Annual State of Agile Development Survey," 2013, p. 4).

Key among the reasons for publishing an Agile guidebook for security professionals are to ensure the success of Agile projects, wherever the reader may find them. According to the surveys from 2012 and 2013, the number one cause of failed Agile projects is a company philosophy or culture "at odds" with core Agile values. Additionally, resistance to change and inability to fit Agile elements into a non-Agile framework are cited as two of the larger barriers to further Agile adoption ("8th Annual State of Agile Development Survey," 2013, p. 5-6), while the lack of perceived planning, documentation, and "predictability" all rate very high in common

concerns for adoption of Agile.

In order to ensure success in Agile projects, the entire team needs to be knowledgeable in Agile development techniques, and believe in their usefulness. Organizational culture and resistance to change may begin with management, but they end with the rank-and-file system administrators, developers, infrastructure, and security engineers to maintain, improve, and develop the system in question. This project seeks to advance knowledge and advocacy for Agile development philosophy as well as provide a workable IT security framework for the potential reader.

Organizational Bias. The problem with VersionOne's survey and results is the inherent organizational bias present in the survey. Indeed, the only real sources of Agile adoption in the enterprise are industry working groups, vendors like VersionOne whose commercial viability depends on the adoption of Agile, and other sources who have an intrinsic bias based on their success riding on Agile adoption. While such bias does not necessarily invalidate the results, it's important to examine this bias. Given the lack of non-agile sources for such adoption statistics, the reliability of the survey can be assessed by taking a look at the survey demographics for clues. Additionally, the presence of other research into Agile methods (including scholarly sources and research around the periphery of the project topic) is indicative of Agile's prevalence, even if that prevalence isn't immediately quantifiable by those sources, providing further evidence of the project's need even before the content is examined.

33% of respondents were classified as "Project managers, team leads, and Scrum Masters ("8th Annual State of Agile Development Survey," 2013, p. 1). These respondents in particular may have a career motive to evangelize for Agile development philosophies, as the industry's use of Agile is experiencing growth (a 4% increase from previous years)("8th Annual State of Agile Development Survey," 2013, p. 2). However, the survey included normal development

staff, HR staff, personnel from administration, marketing, and legal services, as well as support and sales. The majority of the respondents come from non-project management roles. Although there exists organizational bias in the survey results, the inclusion of these people in the survey demographics offsets that somewhat; people in those roles do not have any particular motivation to evangelize for Agile development. Indeed, such people will advocate any method that makes their job easier. Finally, the presence of industry vendors conducting these surveys is indicative of Agile development tools as a profitable endeavor, so the mere existence of surveys like these demonstrate some level of need for the project.

Background on Agile Development and Scrum

Scope of Agile development within the project. The project does not seek to be an all-encompassing handbook on Agile development and Scrum; some assumptions are made as to the reader's background and circumstances. Assuming the reader is an agile developer, practitioner, or project manager who wants a perspective on the information security aspects that IT staff will encounter, the book aims to provide that context through descriptive examples from NBID, and through the examination of topics in information security (focused through an Agile lens). If the reader is an information security professional thrust into an Agile project, the book aims to function as a survival guide, first and foremost: to allow that information security professional to accomplish the mission and adapt to the new, unfamiliar environment. Additionally, the book should provide enough information about Agile methodologies to construct an information security program, without training the reader to be a Scrum Master. For deep immersion in Agile and Scrum, the reader will be referred elsewhere. Ergo, the examination of Agile and Scrum literature is to the goal of taking topics and methods applicable to the security program and distilling the relevant information for the end-reader. To that end, examinations of overarching Agile goals and philosophy, "Sprints" and backlogs, the ubiquitous "stand up" meeting, and team

management topics will be the focus of Agile and Scrum research for the project.

Agile goals and philosophy applied to information security. The Agile Manifesto was written in 2001 by a group of software developers who had taken some existing management and development concepts, and codified it into an overarching philosophy that has become the origin of current Agile software development philosophies and methods. The manifesto does not outline a specific framework or method, but rather outlines the philosophical motivations and goals of the Agile movement:

> We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
>
> - Individuals and interactions over processes and tools
> - Working software over comprehensive documentation
> - Customer collaboration over contract negotiation
> - Responding to change over following a plan
>
> That is, while there is value in the items on the right, we value the items on the left more. (Beck, et al. 2001)

In a process and documentation laden field such as information security, it is easy to see how such a philosophy could cause friction amongst information security professionals. The growth of not only Agile philosophies, but the increased relevance of information security, virtually guarantees this clashing of goals. The challenge, therefore, becomes unifying these goals.

Scrum and other Agile methodologies "strip away non-value-added activities and impel delivery by focusing on the immediate details," with the goal being to "focus on a limited number of tasks within a short planning horizon so we that we can drive them to completion and remove them from our concern" (Pries & Quigley, 2011, ch. 1). Through the lens of a software developer, this means to focus on producing a working product, and eschew comprehensive

documentation or unnecessary focus on processes. However, when viewed through the lens of an information security practitioner, this means something quite different.

Pham & Pham (2012) believe that the current corporate culture has not been reorganized completely for Scrum (ch. 1). They highlight examples from Capital One, Texas Instruments, and other companies that have adopted Scrum, but have kept large swaths of their organizations in place without changes. Nowhere is this more apparent than within the information security field, as the impact of Agile and Scrum's increase in popularity has led to very little change in methodology (as will be examined later in this review). Pham & Pham even admit that there is "much rigor in the traditional approach that even Agile and Scrum can benefit from," and that one of the reasons teams fail with Scrum is that the organization is not set up for Scrum, or that teams do not know how to use Scrum within the framework of the current organization (Pham & Pham, 2012, ch. 1). Indeed, information security is one field where the organizational and philosophical resistance to agile can be significant. First and foremost will be the tendency of any information security officer to recoil in horror at the prioritization of working software over documentation.

In order to overcome this reaction, it becomes necessary to examine the Agile manifesto through the lens of our goals as security practitioners, redefining terms to fit our needs. "Working software over comprehensive documentation" simply means a working, valuable product without unnecessary, exhaustive documentation. It seeks neither to eliminate documentation entirely nor declare it valueless, but trim the scope. "Working software" simply refers to a useful product. In the case of security practitioners, our useful products include our security risk assessments, our controls and mitigations, our monitoring infrastructure, and our IDS/IPS hardware. Risk assessments are critical documentation for the security practitioner, so these are our "working software," for the purposes of Agile methodology, and other

documentation such as lengthy SOPs, detailed vulnerability analyses, acceptable use policies will lessen in importance. The Agile security program will, therefore, place value on critical documentation as a product, lessening the focus on all else. With all the focus on patch management, accreditations and compliance, it can be easy to forget that information security is, at its core, "the process of maintaining an acceptable level of perceived risk" (Bejtlich, 2005, p. 4). The goal of the Agile security program should be to accomplish that mission producing only what is necessary to further that mission, bringing information security squarely into the realm of processes that can be improved by Agile methodology.

Sprints and backlogs as applied to information security. One of the crucial concepts within Scrum is the ordering of tasks for the project. The overall product or project backlog is simply a list of every task intended for the entire project (Pries & Quigley, 2011, ch. 2.3). Obviously, this can be a very large and exhaustive list. Additionally, the backlog is never complete, as it will be continually modified for the duration of the project (Saddington, 2013, p. 36). This concept is easy to apply to information security, with simple collation of existing tasks within the security program into a comprehensive list. In most Scrum implementations, this list includes columns such as an item number, description, duration, responsible parties, priority, advance percentage, and a comments field (Pries & Quigley, 2011, ch. 2.3). The concept of a backlog is also applied to a "sprint," with each sprint having its own backlog.

A sprint is a period of time, defined depending on project that can range from 14 days to a month or more. The sprint's backlog is derived from the project backlog during a sprint planning meeting, and the focus is on getting the tasks completed within the designated timeframe, as there is no such thing as an extension during a sprint (Pries & Quigley, 2011, ch. 2.6). Crucial to the concept of a sprint and the Agile philosophy is the production of "working software," which we have already defined as simply a working product for the purposes of

information security. A team should produce “working software” by the end of a sprint, with “working software” defined as “complete and potentially shippable” (Cohn, 2010, p. 287). In the context of information security, we would define this as artifacts, technology deployments, or infrastructure that is useful immediately, whether for operational use or for preproduction testing. These artifacts could include a draft security risk assessment that, while incomplete, could begin to drive organizational planning for information security. This could also be defined as a pilot on a network security monitoring appliance, which may not yet be scoped to the entire enterprise or project. Incomplete items in a sprint backlog will return to the project backlog at the end of a sprint. The project will also cover information security specifics related to sprints and the planning meetings, including the use of the risk assessment to guide assembly of the sprint. Available literature on Agile development with Scrum is fairly consistent in its delivery of information on sprints and the spring backlog, with some variations on how the sprint planning meeting should be conducted. Constructing an adaptation for information security is simply a matter of defining timelines and terms.

Saddington (2013) describes the sprint planning meeting as producing various action items, including the sprint goal/theme, deliverables, prioritized tasks, and in the case of the initial meeting, a time boxed length that will become standard for future sprints (p. 46). Additionally, before future sprint planning meetings, product owners and managers should be grooming the project backlog, refining and prioritizing it so that the sprint planning meetings are faster and work can begin expeditiously. These principles will be adopted for the project’s Agile information security approach as well, as they represent not only good Scrum practice, but good management practice in general.

Usage of the daily “stand up” in information security. One of the most important tasks in the information security discipline is keeping lines of communication open between team

members. To that end, the crucial daily "stand up," or "Scrum (from which the method is named)" is a powerful tool. The most effective description of the philosophy and goals of the stand-up can be found in Saddington's work:

> Your Agile Team is more effective than any other Tribe out in the jungle because your team members communicate with one another … this stand-up meeting fosters shared accountability; it allows all members to be in tune with the challenges of the day and respond quickly to those changes.
>
> (Saddington, 2013, p. 30)

The primary purpose, therefore, of the Scrum as applied to information security, will be to keep the team informed of each other's activities, and foster teamwork. The meeting is conducted standing up to ensure brevity. Saddington (2013) suggests that any topics longer than a stand up can support should be tabled and taken elsewhere for discussions. Additionally, and this is probably the crucial point that separates a stand-up from traditional meetings, the project team is not there to give the leader an update on his work. The team is there to address the rest of the team about their work, issues, and challenges. (p. 31-32). This fosters shared ownership in the project, and this is directly beneficial to a wholesome information security program.

The stand-up generally follows a format, although it's not a requirement to rigidly adhere to this format. 3 questions are answered by each member of the team:

- What did I do (and complete) yesterday?
- What am I planning to do (and committing to complete) today?
- What impediments do I have that may block my success today?

(Saddington, 2013, p. 31)

Saddington describes the behavior of the leader as crucial to the meeting, while Cohn (2010) mentions the role of the Scrum Master as to "remove impediments to the team's progress (p.

117).” This is important to the role of the leadership in an Agile project versus traditional organizations. The team and product owners establish what needs to be done, and the team gets the job done. The leadership is not directing daily actions; rather they are finding and destroying obstacles on behalf of their team. To that end, the stand-up is a great opportunity for the leadership to plan their activities based on the team members’ responses to the stand-up questions, particularly the last. Both Cohn’s piece and Saddington’s have a similar view of the role of management, and this philosophy is particularly applicable to information security, where most of the work that needs to be done is going to be dictated by standards, law, or policy. This makes the challenge of the Agile security team leader simple: get out of the way, and remove obstacles.

Assuming the Agile security team the project seeks to advise is also working with an Agile development team, the concepts described by Saddington and Cohn should be employed to facilitate cross-attendance of Scrums whenever possible through a Scrum of Scrums. Saddington (2013) describes this as a scrum team made up of representatives from each of several other teams (p. 74). As with anything in Agile, this can be modified to suit the organization, and is conducted precisely the same as any other stand-up. The goal of the Agile security program will be to replace longer, less productive meetings with a short scrum that ends with team members going off to accomplish their work.

Agile team principles. Agile approaches to teams are built on communication and trust. An agile team “has the right to take requirements and specifications and build pieces of the system in the way that they see fit” and autonomy, freedom, and self-organization “brings the team together in unity” (Saddington, 2013, p. 22). Managing team members and human resources is one of the challenges of an information security program, and the Agile method offers numerous ways to build effective teams.

The Agile Manifesto, as discussed previously, values “individuals and interactions over processes and tools” (Beck, et al, 2001). The literature on Agile focuses quite a bit on teamwork, with chapters on these topics in each of the Scrum sources selected. High performance teams trust one another, and in meetings are often seen as “fun, open minded, and caring,” and this behavior is a sign of a team that has built trust, confidence in one another, respect, and know that their jobs are secure (Pham & Pham, 2012, ch. 10). Another way to describe the team (including members, the Scrum Master, and Product Owners) in relation to users, stakeholders, and other outside management, is by using the “pig” and “chicken” categories (Pries & Quigley, 2011, ch. 2.11.3). To wit, if your project is breakfast, the “pigs” are committed, but the “chicken” is only involved. This humorous illustration of the team’s commitment to the project reflects the importance of the Agile team’s ability to function together.

The Agile team’s qualities of accountability, working together, adaptability, collaboration, and communication (Saddington, 2013, p. 22) are essential to the Agile philosophy, and any team across any discipline. Managing these resources as a leader requires commitment to the removal of obstacles, as previously discussed. Indeed, comparisons between all the Agile development literature examined reveal the same “team first, shared responsibility” facets. When a team member is asked who is responsible for a given feature, the answer should be the same regardless of the feature: the team (Cohn, 2010, p. 201). By assembling these principles of teamwork and team management, an information security team can be constructed that trusts one another, shares expertise and avoids silos, and gets the job done in a rapidly changing, difficult environment. The literature on Agile is absolutely unified on the importance of building and maintaining high-performance teams.

State of literature for Agile development. Much has been written on developing using Agile principles. The sources selected represent a cross section, and although there are slight

differences in particular methods, they are surprisingly unified, particularly on subjects useful for information security, including team management, meetings, backlogs and sprints, and the overall philosophy necessary for Agile information security management.

Information Security Principles and Challenges in an Agile Project

Assessing traditional information security methods and practices. Although private organizations, governments, and standards/accreditation bodies all have procedures and processes of their own, the general principles behind an effective security program are unchanging. The over-arching goal of information protection is to "protect an organization's valuable resources, such as information, hardware and software… through appropriate safeguards" in order to "help the organization meet its business objectives" (Peltier, Peltier, & Blackley, 2005, p. 1). Even within organizations with well-defined guidelines, a best practice is to allow different business units "the latitude to make modifications to meet its specific needs (p. 3). Unfortunately, organizational culture can be rigid, and Agile methodologies require flexibility. The challenge, therefore, becomes distilling information security needs to their basic thematic components, and then meeting those low level needs with Agile. Part of this process will be overcoming resistance to change, which the literature examines as well. Resistance to changes in methodology come primarily from technical and social sources, and the social aspect of the change is the source of most resistance (Cohn, 2010, p. 97). The resistance with respect to information security can often be fierce, as the agents of resistance will have security policy and regulations to lean on in support of their resistance. The key elements where Agile process improvements can be made are in risk assessments, implementation of controls, management metrics, and changing organizational culture to return your team to the original purpose of information security.

Risk Assessments. The security risk assessment process has become one of the largest

pieces of information protection programs within the federal government, and some private organizations as well. In the case of the Northern Border Integration Demonstration (NBID) project, it became a detriment to the goals of the project in many cases. In order to successfully conduct security risk assessments in an Agile project, it becomes necessary to return ourselves to the core elements and goals that a risk assessment is intended to meet. Although there are many different formats and frameworks for assessing risk, there exist basic questions asked by all of them:

- What needs to be protected?
- Who/What are the threats and vulnerabilities?
- What are the implications if they were damaged or lost?
- What is the value to the organization?
- What can be done to minimize exposure to the loss or damage?

(Bayne, 2002, para. 3)

In order to do this as efficiently as possible, we must identify through the available research what the core, necessary elements of the assessment are, as well as how to use that information, without bogging down in bureaucracy. The basic elements of a risk assessment are its scope, data collection, analysis of policies and procedures, threat analysis, vulnerability analysis, and correlation and assessment of risk acceptability (Bayne, 2002, para. 8). This document, once created, will assess the value of assets, measure the strength of existing controls and the program, and provides information needed to make planned improvements based on risk (Landoll, 2011, p. 3).

Scope. Scoping a security risk assessment is identified as the most important part of the risk assessment process (Bayne, 2002, para. 6), and the “largest limitation of a security risk assessment is the definition of the boundary being assessed” (Landoll, 2011, p. 52). The

challenge in keeping a risk assessment properly scoped in an Agile project is determining what resources fall within the purview of that project, versus existing organizational infrastructure used as tools, but not necessarily part of the system the development team is responsible for building. For example, Landoll warns against neglecting to include physical security as part of the scope of an organization's security risk assessment (p. 53), and this is generally good advice. However, in the case of many Agile projects, the physical security of a system may be part of a larger organization's security posture, as was the case with NBID, where the Naval Surface Warfare Center – Dahlgren Division owned the building and facilities in which we operated. In the interests of ensuring the security assessment can be produced and updated rapidly, this would be a segment that would only be evaluated with respect to the physical security of the project's individual server infrastructure and rack space. In some cases, a development project may use information technology infrastructure (workstations, networking equipment, and access control systems) for development that the Agile team does not own. In this case, conducting a security assessment (and corresponding penetration tests or scans) could potentially be a breach of policy (Landoll, p. 53). No IT administrator wants to find out a system has been scanned and possibly breached by a team who does not own the infrastructure. Further, over scoping can result in wasted resources and time, a condition unacceptable for rapid development. If the organization has a robust document destruction policy, for example, and the Agile team follows the policy, it would be a waste of resources for the security assessment team to evaluate that policy further if the policy meets overall organizational requirements.

Data Collection. Data collection is the phase by which the security team collects "all policies and procedures currently in place" and begins the process of determining gaps in protection posture (Bayne, 2002, para. 8). This phase, if properly scoped, should produce outputs that are pertinent to the Agile project and only the infrastructure directly related to the

development team's system. The data collected by this phase includes scans that reveal system configuration, patch levels, and even compliance with standards if the correct scanners and plug-ins are used. The data collection phase traditionally also includes collection of policies and current status: administrative, technical, and physical (Landoll, 2011, p. 145). Landoll in particular goes into great detail, enumerating 207 pages of individual configuration and policy points to gather as part of the data collection phase. Although the precise enumeration of each and every possible data point for this phase is beyond the scope of the creative project; the works of Landoll and Brotby both provide an enormous amount of detail for data collection and metrics, which demonstrate aptly the challenge of performing such work in an Agile environment. While the project does not seek to reinvent this work, this has guided the project's discussion on prioritization of security assessment data and threat enumeration. The bulk of the literature, both on security assessments and the later examination of metrics and security posture, focus on completeness. Over scoping is a concern mentioned in many of the works, but the accepted greater danger is in under scoping. This is in steep contrast to the agile methodology, where the prioritization is on a working product (in this case, a security program), with as little extra material as possible.

Analysis of policies and procedures. This portion of the assessment is simply meant to analyze existing policies and gauge organizational compliance (Bayne, 2002, para. 11). The literature differs on whether this kind of analysis is part of the risk assessment, or a separate compliance audit, which does not "measure the security risk to the organizations assets" (Landoll, 2011, p. 13). In this respect, Landoll's otherwise superb work seems to have a weakness: if one assumes that the standards are sound, an audit of compliance to those standards can tell a security team a great deal about the current security posture. It is likely that the difference between Landoll and Bayne in this regard is that Landoll analyzes legal compliance as part of the

risk analysis (2011, p. 378), and seems to base the overall posture of the system on actual threats and vulnerabilities alone, leaving organizational policy out of the equation. Perhaps this "clean slate" approach to risk assessment provides a picture of the organizational posture that is uncolored by existing policies, which may be fallible. To the Agile practitioner, the analysis of policies and procedures will be geared toward examining not only compliance with regulatory guidelines and company policy, but also an examination of whether the company policy supports the mission of the project, and which policies can be ignored or waived.

Threat and vulnerability analysis. Threats are defined as an event with an undesired impact (Landoll, 2011, p. 27), and a threat agent is the entity that could potentially make the threat happen. Bayne (2002) describes a threat as anything that could "contribute to the tampering, destruction, or interruption of any service or item of value" (para. 18). Alternatively, a vulnerability is a flaw in established infrastructure or controls that could allow the threat agent to compromise the system or make the threat a reality (Landoll, 2011, p. 29). Vulnerabilities fit into the overall risk assessment matrix, whichever is used. Commonly, vulnerabilities are rated by potential impact and likelihood, with the results of those ratings combined to form an overall risk. Controls enumerated in the risk assessment require re-evaluation of those vulnerabilities to produce an overall residual risk after controls are applied (Bayne, 2002, para. 17-21).

These analyses must be conducted not only in a thorough fashion, but prioritized based on importance to the project, regulatory or legal requirements, and feasibility of control.

Assessment of Acceptable Risk. The most important concept for the Agile practitioner is understanding and properly communicating the level of acceptable risk. Landoll (2011) reiterates other work in the insistence that there is no such thing as a zero risk environment (p. 33), and it is this concept above all that so many management personnel seem unwilling to accept. The major task related to residual risk that Agile teams must convey is that ultimately the responsibility

rests with leadership to determine actions involved in resolving risk, which can include reducing risk, transferring it to another responsible entity, or accepting the given residual risk (Landoll, 2011, p. 35). Ideally, a cost-benefit analysis should be done on any potential controls, not only in terms of actual financial cost, but manpower and resource requirements. This analysis is done with hardware/software costs, reduction in operational effectiveness, implementation difficulty, staffing costs, and training costs all taken into account (Peltier, et al, 2005, p. 195-197). These costs are then directly measured against the potential cost of a breach, which will include not only financial losses incurred, but costs to reputation and market reaction to a breach. The comparison of costs, combined with the likelihood of the breach, will assist management in determining whether or not the control is operationally and financially feasible.

Information security metrics. Although few security practitioners enjoy collection and analysis of metrics, the effectiveness of a security program cannot be measured without supporting data. In any business, "you cannot manage what you cannot measure" (Brotby, 2009, p. xvii), and this is true in information security as well. As with the enumeration of risk, metrics associated for information security within Agile projects (where timeframe and relevance are key) must be prioritized based on their utility to improving the direct security posture of the environment. The collection of metrics falls into two general categories for an average system: metrics that are easily analyzed without processing, and those that require detailed analysis of not only the metrics, but trends and correlation with other metrics. Patch status and software build versions are examples of the first category, but the second can be much broader. Brotby (2009) enumerates security metrics by both what they measure, and how they are measured (p. 15), with the desirability of objective metrics (such as mean downtime or patch status) being valued over subjective metrics (such as the amount of training required for a user to securely use a service (p. 16). The collection of metrics in an Agile project is limited given the timeframes

between sprints. Metrics which can be highly automated or which are relatively stable over time (and thus, not directly related to a new software build) will be prioritized in any feasible Agile security methodology. In order to ensure the relevance and worth of metrics, an Agile security team must determine the purpose of metrics within their domain. The purpose of metrics of any kind is decision support, so the metrics collected as part of an Agile security program should be entirely chosen based on their use in making decisions, whether those are future software features by the development team, security controls (and effectiveness) implemented by the security team, or even whether the system is operating as expected. When developing metrics, therefore, the first question to be answered is "What kind of decisions are being made within the larger Agile team (to include developers and management/product owners)" (Brotby, 2009, p. 93)? As with the development of the risk assessment, this question involves properly identifying the scope. Although the specific collection and technical methods behind these metrics are beyond the scope of the project book, advice and best practices for selecting and determining metric effectiveness are extremely relevant to Agile security teams, and are covered.

Limitations of existing information security literature. Although the works examined in this section provide expert guidance on traditional security assessment and metrics collection, they do nothing in an effort to apply the concepts within to rapid development projects or Agile methodologies. As such, the information security literature can provide a solid foundation of expected outcomes, but no real guidance toward adaptation. However, given the existing state of Agile literature as applied to development and project management, as well as works applying Agile to other fields covered later in this review, the project goals do not require these works to make that connection. The relevance of this literature to the project remains a foundation for information security practices and the risk assessment deliverable, present in any program. The lack of Agile coverage within the information security literature examined provides another

indication of the project's place within the discipline.

Network Security Monitoring versus Prevention in Agile Projects

Significance of the monitoring literature to the project. The bulk of "The Tao of Network Security Monitoring: Beyond Intrusion Detection" is very technical in nature, with specific intrusion monitoring tools, analysis of TCP dumps, and protocol analyzers comprising the majority of the text. However, the significance of the introductory chapters cannot be understated. The entire book is constructed around a premise that is of immense value to determining the value of security controls and prevention/monitoring efforts; that once you "accept the principle that prevention eventually fails, your worldview changes" (Bejtlich, 2005, p. 13). The core philosophy behind this text is not intended to teach someone to secure a network, but instead to monitor the network, plan for the inevitable breach, and limit the impact of such breaches. This is a departure from not only previously examined texts in this review, but the overall philosophy present within the information security community. While this does not excuse the practice of ignoring prevention, nor does it justify avoiding security controls, this mindset is of great value in determining the direction of an organization's security efforts when time and resources are extremely limited. These limitations are present in Agile projects, with small team sizes, limited budgets, and exceedingly small periods of time for sprints. Concepts in this text pertinent to the project include determining the security status of a system, characteristics of an intruder, the certainty of prevention failure, response versus monitoring and prevention, and exceedingly good guidelines for what constitutes a defensible network.

Are we secure? Contrary to pop culture belief, security is not the complete absence of risk. Bejtlich defines security as "the process of maintaining an acceptable level of perceived risk" (Bejtlich, 2005, p. 4). Often, when a manager or executive asks if a system is secure, they are referring to whether or not that system can be breached. This text not only refutes that definition,

but turns the traditional security mindset upside down. In order to maintain that acceptable level of risk in an Agile environment, the security team must have a realistic expectation regarding their end state and posture. The answer to the question “are we secure?” requires an answer of “I don’t know,” as “no organization can be secure for any time beyond the last verification of adherence to its security policy” (Bejtlich, 2005, p. 4-5). Bejtlich further describes the security process not as a singular journey, but an endless repeating cycle consisting of assessment, protection, detection, and response. Security is not an end state that can be achieved, but rather a continuous, constant process and state of mind. To the Agile practitioner, a repeatable process that can fit within a sprint fits entirely within this unpopular viewpoint, and thus Bejtlich’s text, while far deeper than is necessary for the Agile security team lead, provides valuable perspective that guides the principles by which an Agile security program is constructed.

Characteristics of an intruder. Bejtlich makes assumptions about the nature of a sophisticated attacker that make many network security professionals uncomfortable. Chief among these is the idea that attackers exist that are far more intelligent than the professional attempting to secure the system (Bejtlich, 2005, p. 12). While sophisticated, intelligent attackers are far rarer than a disgruntled teenager running freely available exploits at random, a robust network security program must prepare the organization for the worst case scenario: that an intelligent, sophisticated hacker has targeted the enterprise, and the security program must operate on the assumption that the attacker will eventually succeed.

The second characteristic of an intruder is their unpredictability. The prime limitation of a focus solely on prevention is that the truly sophisticated intruder cannot be predicted. Bejtlich reminds us, “defenders are always playing catch-up… the best intruders save their exploits for the targets that truly matter” (Bejtlich, 2005, p. 12-13). The “zero day” exploit is the exploit that has no current counter. If your project is worth breaching, a sophisticated attacker will use an

exploit not yet in the wild to ensure maximum possibility of success. It is for this reason that Bejtlich advocates a heavy focus on monitoring and response, which is covered below.

Certainty of prevention failure. One of the crucial mindset lessons that the information security practitioner can achieve from Bejtlich's work is the concept that prevention will eventually fail. Having established the uncomfortable fact that some intruders are smarter than the people defending against them, and that they are unpredictable, Bejtlich continues to establish the assumption that your preventative measures will eventually fail, and that while this doesn't justify abandonment of prevention efforts, prevention is not sufficient alone to establish security (Bejtlich, 2005, p. 13-14).

While the task of securing a system (particularly in a rapidly changing Agile environment) remains daunting, the acceptance of these principles drastically alter decision-making process with respect to the expenditure of human and financial resources. Rote adherence to an accreditation document or security implementation guide as used by the Department of Defense no longer seems an effective, holistic security approach. In order to adequately say an organization's security posture is healthy, one must focus more on monitoring and response to threats.

Prevention versus monitoring and response. The eye-opening assertions Bejtlich makes throughout the text bring to light often under-appreciated components of a healthy security posture, and areas in which a time- and resource-limited Agile security team should endeavor to improve: monitoring and response. Monitoring is the "collection, analysis, and escalation of indications and warnings to detect and respond to intrusions" (Bejtlich, 2005, p. 25). While a treatise on monitoring systems and their effective use is beyond the scope of this project, the importance of monitoring with respect to response must be firmly established so that Agile teams can make informed decisions about their infrastructure. The first steps of response are part

of scoping the response. They include:

- What did the intruder do?
- When did he/she do it?
- Does the intruder still have access?
- How bad can this compromise be?

(Bejtlich, 2005, p. 29)

The answers to these questions guide the next steps in the response process, which depend heavily on the nature of the intrusion, and can include disconnecting network servers or segments, changing passwords, temporarily disabling services, and providing forensics to law enforcement to assist prosecution. Key in the response decisions will be the current state of the network with respect to segmentation and design, which are considerations that become part of the characteristics of defensible networks. The challenge for the Agile security team in response is identifying quickly what steps need to be taken, and preferably having a response plan in place prior to a breach.

Characteristics of defensible networks. Bejtlich (2005) describes a number of characteristics of defensible networks. These principles should guide the construction of infrastructure that supports any rapid development project.

- Defensible networks can be watched: a defensible network must be designed with monitoring in mind, allowing engineers to capture traffic, forensics, and information about the operational status of the system, including in-flight connections (p. 20).
- Defensible networks limit an intruder's freedom of movement: network address translation (NAT), segmented networks, and network devices that implicitly allow only needed traffic, denying all others, are among many ways to limit the advance of an intruder who has successfully breached a system (p. 21).

- Defensible networks offer a limited number of services: much as a good permissions policy operates based on the principle of least privilege, a defensible network (or network segment/server) offers precisely the minimum number of services to accomplish the mission, and nothing more (p. 23).
- Defensible networks can be kept current: quite simply, the system's operating system, core and ancillary services, and other software must be kept at the latest patch status whenever possible. If a given piece of custom code does not work with the latest security patch to an operating system (OS) or service, then the code needs to be changed (p. 23-24).

When implementing controls, the Agile security team should ensure that any engineering efforts are focused of furthering a systems adherence to these characteristics.

Limitations of network security monitoring. Like any security strategy, there is no panacea for security posture, and the same holds true for monitoring and response. Collecting every piece of data traversing a network is difficult if not impossible, and analyzing volumes of such data is tougher still. Network security monitoring recommends collecting as much as possible given infrastructure and storage, regardless of ability to analyze it (Bejtlich, 2005, p. 37). Even if the data is not useful real-time, its forensic value post-breach could prove vital. If given the choice, "detection through sampling is better than no detection (Bejtlich, 2005, p. 35). Additionally, network data does not have to be all-inclusive to be valuable. Even traffic analysis consisting of simply "who's talking, for how long, and when" can be valuable if the data is consistently collected and analyzed over time (Bejtlich, 2005, p. 36). The key principle to glean from these limitations for the Agile security practitioner is that resource limitations should never cause abandonment of monitoring altogether; it should merely trigger prioritization of monitoring and analysis choices to allow the monitoring and remediation efforts to fit within the

Agile project's time and personnel limitations. The philosophy on monitoring and prevention espoused by Bejtlich represent a mindset that will serve an Agile security team well when dealing with a rapid development system.

Existing Efforts to Extend Agile

Significance of ancillary efforts to extend Agile to the project. There have been efforts in the discipline to both extend Agile and to describe how to handle common scenarios in an Agile project. Software security, service oriented architecture (SOA), product risk management, and even a paper on Agile IT security will be discussed. The significance of the latter is that it represents an attempt to bridge the gap between IT security and agile development, and the effort will be examined on its merits.

Software security, maintainability, and usability within Agile. In addition to discussing acceptable ways to secure software when developed with Agile methods, an examination of maintainability and usability concerns provide insight into the sustainability of any Agile security program this project hopes to develop. By examining Agile processes through these points of view, it becomes clear that the Agile philosophy is indeed a repeatable, sustainable process that does not preclude security minded decisions.

Ghani, Azham, and Jeong (2014) describe the lack of software security planning built into the Scrum method, and propose the inclusion of a "security backlog" into the method to track emergent security concerns (p. 1-2). The work further describes challenges to Agile software security, including the lack of established security standards that consider agility, the short Scrum release cycle, and the fact that Scrum development teams were not including threat analysis in their requirements capturing (p. 3-4). The Agile IT security method proposed by the project must include considerations for these limitations, as well as absorbing security responsibility that may be overlooked by the development teams. As such, an Agile security

team should have a development resource who can review code. Although the work of Ghani, et al is geared toward the development team, common concepts which will be used to form this project include the employment of use and abuse cases, and the employment of principles found in Microsoft's "Security Development Lifecycle," which is a guide for developing security into software in Agile processes, although the SDL was originally designed for methodologies other than Scrum (p. 5). Although this particular work is oriented strictly toward software development security, these principles are directly applicable in any effort to create an Agile IT security team. Othmane, et al (2014) assert that a new security assessment cannot be performed each time a new iteration is produced within an Agile product. They propose "security reassurance" as a method, allowing the bulk of the assessment to stay in place, with reevaluations of changed modules, automation, and "evidence locality," or the assumption that particular data only applies to parts of the software, and some apply to the entire software package. These same principles can be applied to a holistic IT security plan as well; the Agile security team can determine what components a new build affects or changes, and scope their reassessment based on that.

An additional concern for the development of Agile software (and by extension, attempts to implement an Agile security program) is an evaluation of the sustainability of such an effort. Agarwal and Majumdar (2013) examine the concerns involved, noting that software quality hinges on the maintainability and usability of the software (p. 1-2). Although Agarwal and Majumdar establish the problem, and propose a model for these issues (p. 3), the effort is not mature, and represents very early research. The evaluation of a sustainable IT security process cannot rely on such a model, and will have to rely on existing process improvement frameworks (such as the capability maturity model) for any process improvement concerns.

Agile methodologies applied to service oriented architecture (SOA) development. Although the development of an approach for SOA development using Agile does not

necessarily advance the efforts to unify Agile and IT security, Shahrbanoo, Ali, & Mehran (2012) provide useful insight into a larger problem: unifying two seemingly disparate systems. They state that challenges in unifying SOA with Agile include SOA being a top-down approach, while Agile is a bottom-up development methodology. Additionally, Agile methods do not account for software quality, while SOA does (p. 1-2). The unification method they propose involves prioritization of business goals, the mapping of those goals to processes, extraction of quality scenarios for each goal, determining core business processes, and selecting business processes for the current release (p. 3-5). Similarly, the efforts to unify Agile with information security will require modification of both processes, and creation of new processes to align the two.

Security and risk management using Agile. Although the bulk of the literature actually identifying the topics of IT security comes from academic sources, the existence of the literature suggests that the problem exists, supporting the assertion that this project fills a gap within the market. Indeed, Agile security teams are in play, as is evident by Williams, Meneely, and Shipley (2010), who describe an Agile IT security team and a game they play during iteration planning to assist in enumerating each requirement's relative security risk (p.14). The team discusses the new requirements, and the implications on security risk, and then votes on ease of attack and the value of assets. This method is precisely the construction of a traditional security risk assessment, but with collaborative input and fast procedure (precisely what we would expect from an Agile team). Vaha-Sapila (2010) describes a number of Agile methods to adopt to risk management, including security/user stories and abuse cases, along with the decomposition of large security "epics" into features, then individual tasks (p. 11-13). These methods are hallmarks of agile development, and to apply them to IT security is simply a function of changing terminology. Vaha-Sapila (2010) also challenges organizational resistance, asserting that teams "must have the true power to say no" when asked if the system security is "finished"

(p. 16). Talal Alharbi and Jameel Qureshi (2014) propose the adaptation of Scrum to conducting risk assessments for capability maturity using a model called the "risk register," which is nearly identical to existing composite risk management models, but larger in scope (p. 3). The risks enumerated by this method include such items as a delivery being late, customer feature suggestions, and unclear requirements. It is easy to see how such a method could be used for threat and vulnerability management. Finally, Kesh and Jane (2006) make an effort to apply Agile methods to IT security as a whole, without focusing on any one particular component such as risk management. However, this particular effort is superficial, limited to describing agile methods such as face to face meetings and self-organization, building a team, and the use of decomposition to break a project into smaller tasks (p. 2-3). While the material is correct and sound, it is of insufficient detail and depth to be functionally useful for an IT professional.

Northern Border Integration Demonstration (NBID) Background and Department of Defense (DoD) Concerns

The Northern Border Integration Demonstration (NBID) represents a practical example of the clash between Agile principles and security standards. As NBID will serve as an example illustrating these challenges, background on the project will include news articles in addition to the author's personal experience.

NBID was the centerpiece of infrastructure in the Operational Integration Center at Selfridge Air National Guard base near Detroit, Michigan. NBID represented a demonstration developed through "collaboration of federal agencies… to enhance the ability to detect and deter illegal entry, smuggling activity, human trafficking, and terrorism" (Naval Surface Warfare Center, 2011, para. 2). The facility allows analysts the ability to monitor real time video, radar data, and other technology covering critical border points including 35 miles along the St. Clair River (Cichowski, 2011, para. 2-4). The OIC was created to solve collaboration issues, and bring

more resources to bear on the northern border, an oft-overlooked piece of America's border security challenge. In addition to concerns that militants can transit the northern border without much resistance (Harwood, 2011, para. 1), the Department of Homeland Security stated that in 2010, the Border Patrol only had the resources to respond to incursions at 32 of nearly 4000 miles of the northern border (para 2-3). A great deal of this effort involved the testing and development of solutions in a near-production environment. Many features could not be fully tested without operational use, which introduced challenges to security. Homs (2004) recommends splitting the environment into corporate, engineering, and test/support environments; with the latter being an environment that is co-located in such a way that external users, business partners, and vendors can conduct real-world testing while still maintaining segmentation from critical infrastructure (p. 5-8). At NBID, this hybrid environment was called the "system integration laboratory," but its limitations were clear: there were no operational users, only internal test teams. Additionally, the data feeds for radar and other traffic were provided by simulators; no real world data was used. From a security standpoint, this could have resulted in inability to detect security problems with the delivery mechanisms for real-time data.

Although rigidity within the DoD and federal government were obstacles to NBID, there is evidence mounting that this rigidity is being softened. The National Institute for Standards and Technology (2014) provides a framework for the protection of critical infrastructure data which includes and encourages revisions (p. 15). Indeed, the entire framework published is exceedingly open-ended, more so than would be expected from a government-sponsored framework. The increased reliance on cloud services by the government and DoD reflect a changing mindset as well. Whereas the attitude towards the cloud by government officials used to be a resounding "no," the DoD now looks at ways to leverage this technology as well as properly secure it. The DoD's Cloud Computing strategy specifies the use of cloud services "only if they offer the same

or greater level of protection necessary for DoD mission and information assets" (Edwards, 2014, para. 3). This means that the age old problem of technological and operational fiefdoms within the military is slowly fading away, at least in the information arena, if the DoD is willing to use commercial cloud services over on-premises delivery. This changing mindset within the DoD is precisely the type of change that NBID was borne from.

Summary of the State of Literature

Examination of the body of knowledge within both Agile and information security domains reveals conclusive evidence of a void in which this project fits. Existing efforts to unify Agile development with IT security are largely theoretical and conceptual in nature, with little practical advice for the IT professional who has already been thrust into the situation. Additionally, the state of Agile development literature is mature enough that there is considerable consensus between various sources on methodologies and techniques, providing a solid base for the construction of an Agile information security framework. Through this review, the effectiveness of applying such methods to information security has been established, and the project's place within the discipline seems necessary.

Project Outcome and Findings

Project Journal

Sunday, March 15th, 2015. Throughout the literature review and writing process, this project log will serve as a place to organize my thoughts and discuss relevant questions, along with my progress toward completion. The log will also serve to keep my efforts towards mass-market publication on task. The project has so far expanded a bit in scope, and it may be necessary to reign it in, perhaps scoping the final output to a handbook on security risk assessments in an Agile environment alone, versus the entire security process.

A summary of tasks completed so far:

- Project schedule
- Introduction, project plan, and about half the required sources
- Preliminary Approval from Bob Dibble at NSWC Dahlgren to discuss the NBID project as part of my book.
- Preliminary outline of the book itself.

Today I am reworking the project introduction based on professor feedback. So far, the main concerns are formatting and APA issues.

Tuesday, March 17th, 2015. The research design has been completed, and I move ever forward in the literature review process. The interesting thing about a literature review for this book, versus research for previous academic work, is that the lack of research directly applicable to my topic is actually a positive; it demonstrates the need for my book.

In retrospect, the creative project option has proven to be more difficult, I feel, than a traditional thesis. This is entirely due to its free form nature. With fewer “hard and fast” guidelines, there’s less direction, and a great deal of the effort at this point has been refining the scope and defining my own requirements and success criteria.

Sunday, March 22nd, 2015. Work on the book and literature review marches along, but today I find myself taking a detour to discover the ins and outs of self-publishing via Amazon CreateSpace for both print and eBook formats.

The most important thing is that I get to keep my rights, which I would with both Kindle Direct and CreateSpace. The formatting options are easy, cover layouts can be done via template, and they have semi-formatted Microsoft Word templates for the interior. Their proofing tools are easy, and their costs are minimal. On demand printed copies are only a few dollars each, which is extremely reasonable. I am considering purchasing a nice non-fiction template to jump start the formatting process.

Thursday, March 26th, 2015. The literature review is complete, and layout has begun on the book. I have discovered that the preferred template I'm using has an eBook version available as well, so all my avenues should be covered.

Chapter 3 and 4 are being laid out in the template now. The challenge at this point is walking the line between enough brevity to be a "quick" guide, and enough detail to be operationally useful. The next challenege is how to format the entire project (including introduction, literature review, and all the other academic elements) with the book included. I believe the easiest route is separate files to avoid disrupting pagination and formatting of the book.

Friday, March 27th, 2015. Today marked the adaptation of team principles in Agile development to the information security team. While this is mostly a trivial adaptation due to good team-building techniques being generally industry-agnostic, I did find a comparison that I enjoyed.

Consider users, outside management, and stakeholders. These are "chickens." The project team, Scrum Master, and product owner: These are "pigs." Your project is breakfast. The chickens are only involved. The pigs? Committed.

Wednesday, April 1st, 2015. I was not prepared for just how much thought went into print layout.

Industry conventions, such as justified text, use of footnotes, special formatting to ensure each chapter starts on a right-hand page, and other features require a great deal of work to implement properly.

On a content note, today I received official permission from Dr Eugene Spafford of Purdue University to use a quote of his in the book, across from the first chapter.

Regarding content, I've found that the extensive research and pre-writing process necessary to do the literature review and associated work have resulted in a product that is springing forth very easily. An entire chapter was completed today, including spending several hours experimenting with layout and aesthetics issues.

I have sent the first few chapters to Robert Dibble from NBID for double-checking of the NBID content to ensure I haven't inadvertently included any sensitive material. Additionally, I have sent the first few chapters to my sponsor, Cory Burns.

Sunday, April 5th, 2015. As completion and refining of chapter two continues, I find myself drawn to some reflection on the role leadership has in a healthy agile team.

One of the key principles of agile development is the idea that a leader's primary job is to remove obstacles to the team's success. As I applied this concept to the role of an information security officer, I was reminded of the importance of leadership in maintaining a healthy team, whether the team is using agile principles or not.

If this project is to have utility to the industry, I must continue to drive home the importance of leadership. Perhaps at the end of this chapter when I discuss team-building and conflict resolution within NBID, I will illustrate some positive and negative examples of these leadership traits.

Regardless of a potential reader's current career status, it is reasonable to assume that he or she will eventually be thrust into a leadership role.

This has often been described as a "servant leader," or a leader who builds and nurtures his or her team by serving not only the team's needs, but the needs of the organization. This concept of a selfless leader is present throughout healthy organizations, and should be regarded as the standard by which we measure ourselves as managers.

This is perhaps the biggest impact I can have upon the people who read this book: instilling healthy team-building and trust into these future information security leaders, and constantly reiterating to them the concept of a servant leader.

Saturday, April 11th, 2015. The fourth chapter (of six), is turning out to be more difficult than I imagined. The book layout and chapters have been refactored significantly to reflect a tightening of scope and goals.

Currently, the layout of the chapters and sections is as follows:

1. Introduction
2. The Agile Philosophy for Information Security
3. Scrum methodology for Information Security
4. Agile Security Risk Assessments
5. Compliance in a Rapidly Changing Environment
6. Implementing Controls and Prioritization of Vulnerabilities
7. Conclusion

Additionally, each chapter will have a segment at the end where an anecdote or example from NBID will be examined that is relevant to the previous chapter.

Saturday, April 18th, 2015. The book is nearly 80 pages at the conclusion of the 4th chapter. I estimate around another 40-50 to complete the book. Cover design is in progress, and with the draft project due next week, I've shifted focus to bringing the academic requirements together for turn-in. The book completion, cover, and publication can be done after the draft submission.

I've received fantastic feedback from Bob Dibble, the program manager from NBID. I intend to present this at Agile Alliance conferences, and perhaps at DoD/Public sector conferences as well.

Remaining work includes the final two chapters, finalizing cover design, and ordering a print proof for final revision before publication.

Sunday, April 26th, 2015. The process of preparing a product like this for print includes things like contacting publishers, purchasing an ISBN number, and preparing a web and social media presence. All of these tasks take away time from the writing of the book, and are fairly daunting. I've decided that all ancillary project efforts will remain on hold until the core book is complete. Once the final draft turn in is done, then I will continue to drive layout and marketing tasks and prepare the book for publication.

Saturday, May 2nd, 2015. I have received some valuable notes from Shawn Kolodgie, who was a developer on NBID (and the architect behind some of its most impressive technical features), and an expert on agile. A few inaccuracies about agile methods have been corrected as a result; apparently the "book answer" and the real-world answer can differ in Agile development as well. This project, and the final published book, will be strong thanks to the

comments and feedback from professionals I trust. The compliance chapter is complete, and the final chapter on prioritizing controls is in progress.

Thursday, May 7th 2015. With the exception of proofreading, the book is ready for turn-in. I now shift my focus to the webinar, which will give me an opportunity not only to present my work, but hopefully advocate for the creative project option. I believe this option, for information security professionals, to be a more valuable experience and more relevant to the workforce, due to the free form nature and self-imposed success criteria.

Friday, May 15th, 2015. Another round of feedback from my sponsor and other professionals has resulted in some minor content alterations to the book. As I prepare for the project webinar, I feel that within the context of my program here at AMU, this project is now complete. Although there remains significant refinement before publication, the project goals and acceptance criteria were met. Further editing, refinement, and cover design remain the only tasks to complete before the book goes to press.

Appraisal of the Project

Assessment of the project acceptance criteria. The project criteria indicated the following goals, which have been accomplished:

- Provide the reader an acceptable background on Scrum and Agile development.
    - Using Scrum and agile principles identified through the literature review, a solid foundation of not only Agile methods, but also Agile philosophy and leadership principles was provided in the first two chapters. The third not only described the Scrum process to the reader in detail, but applied it directly to common information security tasks.
- Describe the NBID project in enough detail to be relevant, but meet with NBID PM release approval

    - No sensitive information or restricted information has been identified by NBID PM Bob Dibble, and the anecdotes from NBID at the end of each chapter illustrate the concepts previously discussed.
- Adequately provide the reader sound guidelines for information security activities in an agile environment
    - Using established best practices from the literature review, a workable agile security program framework is described in detail by the book. The book successfully returns to the basics of sound information protection.
- Be ready to publish in traditional industry non-fiction book format.
    - The book is entirely print ready, with an industry standard non-fiction layout, and conforms to stylistic and structure conventions found throughout the discipline.

Findings and assessment of the project. Although the final effectiveness of the book will not be known until publication and expansion to a wider audience, some conclusions can be drawn based on not only the writing process, but also feedback from my project sponsor. As demonstrated in the literature review, there exist significant problems with securing systems developed with agile methodologies, due to the rapid nature of change. Even within a relatively static system, the threat landscape moves quickly, and it can be a struggle to keep up. The existing efforts to unify agile principles with information protection are not mature, and therefore this project has the potential to be a significant contribution to the discipline.

The original question posed at the genesis of the project has been satisfactorily answered: A framework for Agile Information Security has been constructed which meets the following parameters:

- The method presented by the book is flexible enough to be modified to various situations and projects, containing guidance on how to mold the method to fit

organizational needs, and with a construction that allows agile improvements to be implemented piecewise.

- Further, the method has enough rigor to provide solid, useful guidance to IT professionals. This rigor is achieved through adherence to best practices and a complete chapter on compliance and verification concerns using Agile methods.

Further, the book focuses on a well-needed "return to basics" approach regarding information protection, on a philosophical level. The very mindset of information protection supporting the mission, instead of becoming the mission, has immense value within the industry. The project required careful analysis and use of concepts from the entire program of study in the information security concentration, and this program of study, coupled with the project research itself, have resulted in a book that promotes well-established best practices from both Agile development and information protection in a unified, novel way. Above all, a personal goal was met: the book provides sound guidance and help to any information technology professional who is experiencing the same challenges the NBID team experienced; securing a rapidly changing system with dwindling resources and ever-increasing requirements.

References

7th Annual State of Agile Development Survey. (2012). http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf

8th Annual State of Agile Development Survey. (2013). http://www.versionone.com/pdf/2013-state-of-agile-survey.pdf

Agarwal, M., & Majumdar, R. (2013). Software maintainability and usability in agile environment. *International Journal of Computer Applications*, 68(4) doi:10.5120/115696873

Bayne, James. (2002). An overview of threat and risk assessment. *SANS InfoSEC Reading Room*. http://www.sans.org/reading-room/whitepapers/auditing/overview-threat-risk-assessment-76

Bejtlich, R. (2005). *The Tao of network security monitoring: Beyond intrusion detection*. Boston: Addison-Wesley.

Beck, Kent, et al. (2001, February). The Agile Manifesto. *Agile Alliance*. http://www.agilealliance.org/the-alliance/the-agile-manifesto/

Brotby, W. (2009). *Information security management metrics: A definitive guide to effective security monitoring and measurement*. Boca Raton: CRC Press.

Cichowski, M. (2011, March 24). Northern border security goes high tech. http://liveshots.blogs.foxnews.com/2011/03/24/northern-border-security-goes-high-tech/

Cohn, M. (2010). *Succeeding with Agile: Software development using Scrum*. Upper Saddle River, NJ: Addison-Wesley.

Edwards, J. (2014). DoD's cloud security challenge. *C4ISR*, 29. Retrieved from http://search.proquest.com/docview/1511405642?accountid=8289

Ghani, I., Azham, Z., & Jeong, S. R. (2014). *Integrating software security into agile-scrum*

method. KSII Transactions on Internet and Information Systems(TIIS), 8(2), 646663.doi:10.3837/tiis.2014.02.019

Harwood, M. (2011, December 1). North by northwest. *Security Management*. https://sm.asisonline.org/migration/Pages/north-northwest-009248.aspx

Homs, A. (2004, September). Internal security in a engineering development environment. *SANS InfoSEC Reading Room*. http://www.sans.org/reading-room/whitepapers/bestprac/internal-security-engineering-development-environment-1511

Kesh, S., & Jane, S. (2006). Applying agile methodologies to IT security. *Issues in Information Systems*, 7(2), 73-76.

Landoll, D. (2011). *The security risk assessment handbook: A practical guide for performing security risk assessments* (2nd ed.). Boca Raton, Fla.: CRC Press.

National Institute of Standards and Technology. (2014). Framework for improving critical infrastructure cybersecurity. http://www.nist.gov/cyberframework/upload/cybersecurity-framework-021214.pdf

Naval Surface Warfare Center – Dahlgren Division. (2011). Navy engineers and scientists support new Customs and Border Protection Operational Integration Center. http://www.navsea.navy.mil/nswc/dahlgren/NEWS/NBID/NBID.aspx

Othmane, L., Angin, P., Weffers, H., & Bhargava, B. (2014). Extending the agile development process to develop acceptably secure software. *IEEE Transactions on Dependable and Secure Computing*, 11(6), 497509. doi:10.1109/TDSC.2014.2298011

Peltier, T., Peltier, J., & Blackley, J. (2005). *Information security fundamentals*. Boca Raton, Fla.: Auerbach Publications.

Pham, A., & Pham, P. (2012). *Scrum in action: Agile software project management and development*. Boston: Course Technology.

Pries, K. H., & Quigley, J. M. (2011). *Scrum project management*. Boca Raton, FL: CRC Press.

Shahrbanoo, M., Ali, M., & Mehran, M. (2012). An approach for agile SOA development using agile principals. *International Journal of Computer Science & Information Technology*, 4, 237244.

Saddington, P. (2013). *The agile pocket guide: A quick start guide to making your business agile using scrum and beyond*. Hoboken, New Jersey: John Wiley & Sons.

Talal Alharbi, E., Jameel Qureshi, M. R. (2014). Implementation of risk management with SCRUM to achieve CMMI requirements. *International Journal of Computer Network and Information Security*, 6(11), 2025. doi:10.5815/ijcnis.2014.11.03

Vaha-Sipila, A. (2010). Product security risk management in agile product management. *Open Web Application Security Project.* https://www.owasp.org/images/c/c6/OWASP_AppSec_Research_2010_Agile_Prod_Sec_Mgmt_by_Vaha-Sipila.pdf

Williams, L., Meneely, A., & Shipley, G. (2010). Protection poker: The new software security "game". *IEEE Security&Privacy Magazine*, 7(3), 1420. doi:10.1109/MSP.2010.58

School of Science, Technology, Engineering, and Math

Information Technology

The thesis for the master's degree submitted by

James Richard Fitzer

under the title

Agile Information Security Using Scrum

has been read by the undersigned. It is hereby recommended

for acceptance by the faculty with credit to the amount of

3 semester hours.

(Signed, first reader) **Novadean Watson-Stone** (Date) **May 22, 2015**

(Signed, second reader, if required) _____ (Date) _____

Recommended for approval on behalf of the program

(Signed) _____ (Date) _____

Recommendation accepted on behalf of the

program director

(Signed) _____ (Date) _____

Approved by academic dean

Date: May 22, 2015

I, **James Richard Fitzer**, owner of the copyright to the work known as "Agile Information Security Using Scrum," and the associated book, "Agile Information Security: Using Scrum to Survive In and Secure a Rapidly Changing Environment, hereby authorize **APUS** to use the following material as part of his/her thesis to be submitted to American Public University System.

Page Line Numbers or Other Identification

_____

Signature

Once the signatures have been secured, the supervising professor or program director should complete the form below and include it when submitting the final document to the APUS Online library.

Submission Information—email: ThesisCapstoneSubmission@apus.edu

This capstone has been approved by (**Novadean Watson-Stone**) for submission, review, and publication by the Online Library. (The thesis advisor must note whether the thesis is accepted generally or accepted with distinction.)

Author's name: James Richard Fitzer

Title: Agile Information Security Using Scrum

Professor: Novadean Watson-Stone

Second reader, if required: N/A

Program: Information Technology

Pass with Distinction:

YES NO

Keywords/Descriptive Terms: **Agile development, information security, information protection, Scrum**

[ ] Contains Security-Sensitive Information