Cover Sheet
Paper #105

20th ICCRTS

Title of Paper:
Introduction to Agile Security Concepts and Fundamentals, with C2 Implications

Primary Topic:
Topic 11: Agile C2 Security

Alternates Topics
Topic 2: Organizational Concepts and Approaches
Topic 12: ISR for Decision Making

Name of Author
Rick Dove

Point of Contact (POC) who will be responsible for all correspondence with the ICCRTS team.
Rick Dove
Paradigm Shift International
2051 Taos County Road B-014, Box 289, Questa, New Mexico 87556
+1-575-586-1536
dove@parshift.com

Abstract: This paper defines agile security as both a resilient (reactive) and composable (proactive) capability to effectively address the unpredictable and evolving environment of adversarial attack – a systems engineered capability enabled by systems architecture and design. The justification of that definition is supported with history and work that discovered fundamental architecture and design principles which enable agility in systems of any kind. These fundamentals are then applied as notional concepts for guiding the development of agile security systems, with an emphasis on cyberphysical and C2 systems. Shortfalls in current thinking toward fully effective agile security are than supported with suggestions for consideration.

Introduction to Agile Security Concepts and Fundamentals, with C2 Implications

Rick Dove, Paradigm Shift International, rick.dove@parshift.com

Abstract: This paper defines agile security as both a resilient (reactive) and composable (proactive) capability to effectively address the unpredictable and evolving environment of adversarial attack – a systems engineered capability enabled by systems architecture and design. The justification of that definition is supported with history and work that discovered fundamental architecture and design principles which enable agility in systems of any kind. These fundamentals are then applied as notional concepts for guiding the development of agile security systems, with an emphasis on cyberphysical and C2 systems. Shortfalls in current thinking toward fully effective agile security are than supported with suggestions for consideration.

Introduction

In today's ubiquitous cyber environment all systems are prey, but some are more critical than others. Infiltrating a drone can cause some damage, but hacking a drone command and control (C2) network can disable an entire war fighting capability. Of course C2 encompasses much more than drone fleets, whether military, industrial, or otherwise. Agile security is an architectural design concept that enables both reactive resilience and proactive composability, providing the possibility of at least parity with the agile adversary. This paper will briefly review current work in agile security focused on reactive resilience, and the current call for proactive composable security; and then review agile architectural design fundamentals for any system that needs to operate in an uncertain, unpredictable, evolutionary environment. Suggestions for consideration in current thinking will then be outlined.

The adversarial community readily shares and mines strategies and tactics, leading evolution and innovation, whether nations state backed or not. It is time to rectify this. "Cyberspace is becoming a highly asymmetric environment where even small groups or individuals can operate with some degree of effectiveness against even large organizations by a variety of means. Many of these small groups share information and operational knowledge through user networks and thus have larger effective footprints and resources." (Kadtke 2014:24).

Agile security is addressed here as a system's engineered capability which enables agile systemic behavior. The word agile, uncapitalized, is used as an adjective. When both words are capitalized, as in *Agile Security*, something all together different is present, in the nature of a product brand (Sourcefire n.d.) with presumed market appeal, but of no relevance to this discussion.

Here, agile security is defined as both a resilient (reactive) and composable (proactive) capability to effectively address the unpredictable and evolving environment of adversarial attack – a systems engineered capability enabled by systems architecture and design. This definition is agnostic to the type of system, and can encompass information systems, cyberphysical systems, physical systems, enterprise systems, social systems, and military systems.

Agile security is not security developed by an agile software development process, as presumed by (Konda n.d.) and some others. Agile security is the product of a systems engineering effort independent of the process employed in development.

A little history of agile concepts will provide some context. Agile systemic behavior was defined in 1991 as the ability to respond effectively and with competence to operational environments that exhibit uncertainty and unpredictability (Dove 1991, Dove 1992). That ability was named agility, and that study spawned the Agility Forum (nee Agile Manufacturing Enterprise Forum) to explore, throughout most of the '90s, the nature of agile enterprise and domain-independent agile systems. The primary focus of the 1991 study was on the agile manufacturing enterprise, as a likely successor to the lean manufacturing enterprise. The work in process was socialized widely for feedback with groups such as NIST, DARPA, the Defense Science Board, the Aerospace Industries Association, and others recognized in (Dove and Nagel 1991); which likely sparked subsequent military "agile enterprise" interests such as force transformation (Cebrowski 2003). The subsequent Agility Forum study broadened the focus to agile systems of all kinds, and began the search and development of agile-system enabling fundamentals.

With the agile label and concept in play, Hewlett Packard was the first to initiate a program to educate its customers (Dove 1993) and subsequently bring to market IT support under the Agile Enterprise banner; DoD's Command and Control Research Program began an exploration of agile command and control (Alberts 1996), which continues today with a broadened conceptual migration into agile military enterprise (Alberts 2011); and the Agile Manifesto for Software Development (Fowler 2001) adopted the agile label as appropriately descriptive and fundamentally consistent with their software development concepts[1].

Most recently, the International Council on Systems Engineering (INCOSE) has elevated agile systems and agile systems engineering to one of its top five Corporate Advisory Board priorities (Dove 2015), and has made these priorities, as well as agile security, a key part of its Vision 2025 (INCOSE 2014). INCOSE working groups on Agile Systems and Systems Engineering and on Systems Security Engineering, both chaired by this author, are

[1] Personal communication with Jim Highsmith, a founder of the Agile Manifesto for Software Development (Fowler and Highsmith 2001).

collaborating on the nature of agile security from a systems engineering perspective. For the first time, security engineering will be included in the INCOSE Systems Engineering Handbook, to appear in Version Four scheduled for publication in first quarter 2015.

Systems engineering, rather than security engineering, is the necessary initial focus for agile security, and the focus of this introductory paper. For instance, resilient systems are resilient first because of a general systems architecture and design, which may have nothing to do with security devices or security applications, per se.

From the systems engineering standards point of view (ISO/IEC/IEEE 2008), security is classed as a specialty engineering element rather than a functional engineering element, and as a result is often postponed in the system engineering activity until after system architecture, design, and development is mostly, if not fully, completed. Perhaps worse yet, system security is often implemented simply as conformance to specified standards and regulations to satisfy stated contract requirements and so-called best practices.

Nature takes a different point of view, predicated on the fact that an organism is non-functional and extinguished as a species if security isn't the initial functional concern. Living another day is the functional prerequisite of all higher functional performance.

This paper will discuss agile security from a systems engineering perspective: showing fundamental agility-enabling system architecture and design concepts, relating those concepts to systems security in general and command and control in particular, and suggesting some initial working principles for effective agile security.

Defining agile systems capability (Dove 2014):

> Agility is the ability of a system to thrive in an uncertain and unpredictably evolving environment; deploying effective response to both opportunity and threat, within mission. Effective response has four metrics: timely (fast enough to deliver value), affordable (at a cost that can be repeated as often as necessary), predictable (process that can be counted on to meet the need), and comprehensive (anything and everything within the system mission boundary).

Agile systems have effective situational response options, within mission, under:

- Unpredictability: randomness among unknowable possibilities.
- Uncertainty: randomness among known possibilities with unknowable probabilities.
- Risk: randomness among known possibilities with knowable probabilities.
- Variation: randomness among knowable variables and knowable variance ranges.
- Evolution: gradual (relatively) successive developments.

Review of Agile Security Work-in-Process

To set context, this section provides an overview of current work and thinking about security concepts that exhibit agile capability, occurring under a variety of different labels. The intention is to show that these various labels are encompassed rightfully under the agile security label. Six terms are chosen for examination, that cover the ground of interest in this paper, quoting principle players as they define and explain the terms.

Related terms of interest:

- Adaptive
- Resilient
- Composable
- Active
- Agile
- Autonomic

Adaptive – From (Emami-Tabla 2013): "Systems that feature adaptive security detect and mitigate security threats at runtime with little or no administrator involvement. In these systems, decisions at runtime are balanced according to quality and performance goals. … *Adaptive security* refers to solutions that aim to adapt their defence mechanisms at runtime." Adaptive security is relatively limited agile-security concept, appearing to focus its application in advanced Intrusion Detection Systems and methods.

Resilient – From (DoD 2011:6) "Operating with a presumption of breach will require DoD to be agile and resilient, focusing its efforts on mission assurance and the preservation of critical operating capability. These efforts will be supported by the development of increasingly resilient networks and systems. In the case of a contingency involving network failure or significant compromise, DoD must be able to remain operationally effective by isolating and neutralizing the impact, using redundant capacity, or shifting its operations from one system to another. Multiple networks can add diversity, resiliency, and mission assurance to cyberspace operations. … DoD will continue to be adaptive in its cyberspace efforts, embracing both evolutionary and rapid change."

From (Bodeau 2011:8), under Working Definitions:"*Cyber resiliency is* The ability of a nation, organization, or mission or business process to anticipate, withstand, recover from, and evolve to improve capabilities in the face of, adverse conditions, stresses, or attacks on the supporting cyber resources it needs to function." Later, on page 14, under Cyber Resiliency Engineering Framewor: "The elements of cyber resiliency consist of four goals (Anticipate, Withstand, Recover, and Evolve), eight objectives (Understand, Prepare, Prevent, Continue, Constrain, Reconstitute, Transform,

and Re-architect), and an extensible set of (currently fourteen) practices (Adaptive Response, Privilege Restriction, Deception, Diversity, Substantiated Integrity, Coordinated Defense, Analytic Monitoring, Non-persistence, Dynamic Positioning, Redundancy, Segmentation, Unpredictability, Dynamic Representation, and Realignment) that are intended to maximize cyber resiliency. The various goals, objectives and practices do not stand in isolation. For example, unpredictability (a practice) can play a key role in achieving effective deception (another practice)."

From (Collier 2014): "Assessment mechanisms must continuously assimilate new information and track changing stakeholder priorities and adversarial capabilities. Successful implementation of these mechanisms will depend on the integration of risk- and resilience-based management in an adaptive framework. … Cyberthreats' dynamic nature is one reason for the emerging concept of cyberresilience. Unlike risk assessment, which focuses on detecting and preventing a known threat, cyberresilience focuses on the ability to prepare for and recover quickly from both known and unknown threats. Risk management is concerned primarily with hardening a system component to avoid its failure; resilience management is concerned with maintaining the system's critical functionality by preparing for adverse events, absorbing stress, recovering the critical functionality, and adapting to future threats."

From (DHS 2014): "The Enterprise Automated Security Environment (EASE) concept envisions an environment in which automated and dynamic enterprise-level cyberspace1 defense capabilities such as adaptive sensing, sense-making, decision-making, and acting provide shared situational awareness and support response in cyber-relevant time. … Enhance the abilities of stakeholders to rapidly and securely exchange and appropriately share actionable data and information about threats, attacks, adversary techniques, etc., to establish situational awareness and improve response. … Develop and realize a modular, plug-and-play environment for cyberspace defense of .gov and .mil that supports the following objectives: rapid insertion of new technologies from diverse existing and emerging vendors; inter- and intraorganizational communications; effective testing of new capabilities; and effective policy implementation.

Composable – From (Savage 2010): "Composable Capability on Demand … [as a] concept represents a deliberate shift away from traditional C2 systems acquisition and development strategies toward developing and fielding a flexible infrastructure and separate components that can be rapidly integrated to create a new and enhanced capability that a situation demands. With this approach, users will be able to respond to situations—whether it's setting up a command center on the battlefield or during a hurricane—with an adaptable, interoperable capability.

Figure 2. CCOD Evaluation Criteria

- The function of a *resource* can be rapidly and easily understood by a user.
- *Resources* can be determined to be trustworthy and secure.
- Users at various skill levels can effectively compose resources into capabilities.
- Elements of composition can rapidly be brought into semantic alignment to allow their execution as part of a composition.
- Compositions can be republished as reusable functions.
- Ability to identify and respond to breakage in a composed capability.
- *Resource* or composition can be dynamically scaled within minutes in response to demand.
- CCOD functions can be effectively executed across a span of infrastructure environment.
- Multiple simultaneous *resource* "storefronts" can function (i.e. deliver and/or execute) in harmony.
- A competitive and rapid solicitation/development market for resources can be established. Network operational parameters and interactions are sufficiently dynamic to support *resource* provisioning.

Resource: e.g. Application, Service, Widget

This work focuses on acquisition of C2 functional capability, and does not call out security capability specifically. Nevertheless, this author (of paper you are reading) views system security as a functional need, contrary to the current ISO/IEC/IEEE 15288 classification of security as a specialty-engineering non-functional element. In any event, the Figure 2 Evaluation Criteria above, proposed by (Savage 2010), is a good notional framework for defining composable capability, in need of only a little work to be more general and appropriate for agile security.

Active – From (DoD 2011:7): "Active cyber defense [ACD] is DoD's synchronized, real-time capability to discover, detect, analyze, and mitigate threats and vulnerabilities. It builds on traditional approaches to defending DoD networks and systems, supplementing best practices with new operating concepts. It operates at network speed by using sensors, software, and intelligence to detect and stop malicious activity before it can affect DoD networks and systems."

From (Herring 2014): "ACD is not a single solution; it is a capability to provide context and interoperability among many solutions under the six functional areas. An integrated, cohesive ACD solution implies the use of many sensors, analytics, and displays to support many decision-makers. For example, ACD may accommodate any number of analytics from any number of perspectives. The type and focus of the analytic is dependent upon the needs of the decision-maker who will use the results of the analysis. ACD intends to accommodate what is available today (current tools) and what will be available tomorrow (future tools yet unknown). This leaves room for new, better, faster, and cheaper solutions across all functional areas.

From (Willett 2015b) "Fundamental design principles behind IACD [Integrated Active Cyber Defense] are to create a layered and modular design. This implies highly specialized components that perform discreet tasks that take input from other components and provide output to other components. Such a design is ultimately more flexible (i.e.,

dynamic, adaptable, agile) for discreet refinement, improvement, replacement without the need for broad systemic modifications. … The grand vision for IACD culminates with the realization of cybersecurity automation. Cybersecurity automation has two macro layers: 1) management for making decisions, and 2) mechanistic manipulation for taking action. Mechanistic manipulation consists of the ability to modify the cyberspace environment on-the-fly including the infrastructure, desktops, servers, cybersecurity services and mechanisms, etc. The management layer consists of governance functions that decide (in its literal sense) what to do and adjudication functions that identify problems and decide (in its literal sense) how to resolve them."

Agile – From (Dove 2005a): "*Agile-Response Framework* - By expressing the requirements of a strategy, and each of its constituent elements, in terms of a fitness function, proposed solutions can be filtered for acceptability before specific features are weighed against competing alternatives. This framework is an initial strawman adapted generally from agility research and subsequently augmented specifically for an agile-security-strategy working paper at a semiconductor foundry. This is an untested framework that is expected to undergo considerable refinement and augmentation during the project.

Reactive Principles

1. Detection – Detect intrusion and damage quickly
2. Containment – Minimize potential damage scope
3. Mitigation – Minimize potential damage magnitude
4. Assessment – Understand what has been damaged and how
5. Recovery – Repair damage quickly
6. Accountability (Reactive) – Identify the perpetrators forensically, after damage

Proactive Principles

1. Vulnerability/Risk Anticipation – Identify pending changes to vulnerability and risk before occurrence
2. Prudence – Correct vulnerabilities before exploitation, sense indirect indicators of pending exploitation
3. Transformation – Change randomly the elements/nature of security system
4. Threat Anticipation – Identify and counter threats and risks before exploitation
5. Migration – Continuous upgrade of security strategy and components
6. Accountability (Proactive) – Identify perpetrators with traps, glass houses, disinformation, etc, before damage

From (Dove 2009): Current-generation security is characterized principally as reactive: it is invented and deployed in response to the escalating sophistication of attack experiences. As an after-the-fact defense insertion, it is typically an add-on functional subsystem, force-fit to the system that needs protection. In contrast, next-generation security is an emergent property of the system it protects. To provide parity with the agility of intelligent attacking systems, six characteristics are needed:

- self-organizing,
- adapting to unpredictable situations,
- evolving in concert with an ever changing environment,
- reactively resilient,
- proactively innovative, and
- harmonious with system purpose.

From (Dove 2013): "Fielding sustainably secure systems today is critical to system mission needs, yet difficult when system security is less than a paramount thoughtful concern of the system engineering processes. Responsibility lies with both acquirer, to demand it, and supplier, to provide it even when not demanded. The acquirer must place responsibility for system security on the systems engineering activity. The supplier must enable sustainable security and enable agile lifecycle security processes throughout the operational lifetime."

From (Willett 2015b): "Agile security is the well-coordinated ability to move quickly and easily in order to maintain an acceptable state of exposure to harm or danger. To be secure is not a goal, it is a state of being. The measure of being secure is not absolute, but is a factor of many continually changing influences including threats, assets, vulnerabilities, risk, and risk tolerance."

From (AFRL 2014), Command and Control of Proactive Defense (C2PD): "Cyber agility techniques called moving target defenses offer a capability to assure the network and Air Force missions. By providing mobility to static network and computing resources within the enterprise, we create uncertainty for the attacker and can outmaneuver attacks to critical cyber infrastructure. However, without a command and control structure to plan, assess, and execute a coordinated defense, we may expose a larger attack surface to the network and increase the risk of cyber fratricide."

INCOSE's Vision 2025 document (INCOSE 2014) does not call out agile security as a labeled category, but separately calls for security-encompassing composable systems and resilient systems. Envisioning the objectives: "Systems engineering routinely incorporates requirements to enhance systems and information security and resiliency to cyber threats early, and is able to verify the cyber defense capabilities over the full system life cycle, based on an increasing body of strategies, tools and methods. Cyber security is a fundamental system attribute that systems engineers understand and incorporate into designs. … Systems engineering practices will deal with systems in a dynamically changing and fully interconnect systems of systems context. Architecture design and analysis practices will enable integration of diverse stakeholder viewpoints to create more evolvable systems. Design drivers such as cyber-security

considerations and resilience will be built in to the solution from the beginning. Composable design methods will leverage reuse and validated patterns to configure and integrate components into system solutions."

Autonomic – Encompassing agile security, autonomic computing is a grand vision, with serious active work already claiming implemented progress, an *International Journal of Autonomic Computing*, and eleven annual conferences through 2014.

From (IBM 2001): "It's time to design and build computing systems capable of running themselves, adjusting to varying circumstances, and preparing their resources to handle most efficiently the workloads we put upon them. These autonomic systems must anticipate needs and allow users to concentrate on what they want to accomplish rather than figuring how to rig the computing systems to get them there. … In the end, its individual layers and components must contribute to a system that itself functions well without our regular interference to provide a simplified user experience. Such a high-level system could be described as possessing at least eight key elements or characteristics.

1. *To be* AUTONOMIC, a computing system needs to "know itself "—and comprise components that also possess a system identity.
2. *An* AUTONOMIC COMPUTING SYSTEM must configure and reconfigure itself under varying and unpredictable conditions.
3. *An* AUTONOMIC COMPUTING SYSTEM never settles for the status quo — it always looks for ways to optimize its workings.
4. *An* AUTONOMIC COMPUTING SYSTEM must perform something akin to healing — it must be able to recover from routine and extraordinary events that might cause some of its parts to malfunction.
5. *A* VIRTUAL WORLD is no less dangerous than the physical one, so an autonomic computing system must be an expert in self-protection.
6. *An* AUTONOMIC COMPUTING SYSTEM knows its environment and the context surrounding its activity, and acts accordingly.
7. *An* AUTONOMIC COMPUTING SYSTEM cannot exist in a hermetic environment.
8. *Perhaps* most critical for the user, an autonomic computing system will anticipate the optimized resources needed while keeping its complexity hidden.

From (Kephart 2003): "The term autonomic computing is emblematic of a vast and somewhat tangled hierarchy of natural self-governing systems, many of which consist of myriad interacting, self-governing components that in turn comprise large numbers of interacting, autonomous, self-governing components at the next level down. The enormous range in scale, starting with molecular machines within cells and extending to human markets, societies, and the entire world socioeconomy, mirrors that of computing systems, which run from individual devices to the entire Internet. Thus, we believe it will be profitable to seek inspiration in the self-governance of social and economic systems as well as purely biological ones."

Engineered Agile Systems - Fundamentals

The fundamental needs and concepts of agile systems capability were developed throughout the nineties in two industry-collaborative projects led by Lehigh University and funded by the US Department of Defense (DoD) (Dove 2001). The projects involved over 1,000 people from all types of organizations, and analyzed real systems that exhibited agile capabilities. Out of this emerged a fundamental common architecture and set of design principles that enable agile capability in any system domain (Dove 2014).

Two of the terms discussed earlier warrant key attention: resilient and composable. Since the early agile systems work in the '90s, variations on the quad graphic of Figure 1 have been used to make the point that agility is composed of both reactive and proactive change proficiency. Since then, resilient systems have become a strong focus of interest and study, and more recently a call for composable systems is being heard. In both cases an ability to reconfigure system resources effectively to deal with new environmental situations is called for. As will be shown later, this ability to change effectively is enabled by a fundamental architecture common to both.

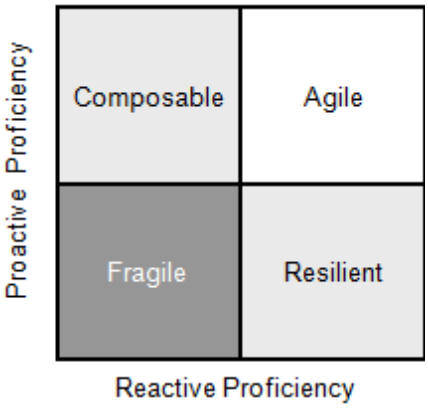

Figure 1. Two dimensions of response proficiency

Relating resilience to agility: Consolidating some 15 years of agile command and control investigation for the US DoD, Dave Alberts (Alberts 1996:108) recognizes resilience as one of six components (his word) of agile systems; and juxtaposes the capability with a need to respond to a "Change in Circumstances: The destruction, interruption, or degradation of an entity capability. … Resilience provides an entity with the ability to repair, replace, patch, or otherwise reconstitute lost capability or performance (and hence effectiveness), at least in part and over time, from misfortune, damage, or a destabilizing perturbation in the environment (Alberts 2011: 217-218)."

Relating composability to agility: In a recent paper addressing military "composable capability", Hillary Sillitto proposed: "…improved operational readiness, performance and interoperability can be achieved by applying a systems

engineering methodology in which the 'system focus' is the force element, not the individual equipment; it is possible to identify a finite set of stable, well characterised building blocks (Force Elements) from which a wide variety of task force structures can be put together, providing almost infinite variety of capability solutions; …" Sillitto suggests that the "System Coupling Model" (Lawson, 2010:23) sets the context of "composability," reproduced in Figure 2 as a condensed version of the agile architecture pattern shown in Figure 3.

Resilient systems are themselves a security concept that gets implemented without the need for dedicated security technology. This is a systems engineering architectural concept that results in a (more) secure system. Resilient systems are a shift away from the specifics-based risk and vulnerability analysis as a solution driver, to a general concept that doesn't care where the threat comes from, or when, or by who/what, and focuses instead on sustaining system functionality.

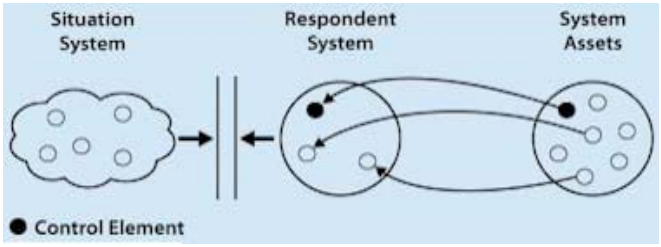

Figure 2. System-Coupling Diagram (Lawson 2010:23) illustrating composability of a response system appropriate to a situation.

Effective response to both opportunity and threat is depicted in Figure 1 as response proficiency in two dimensions: proactive and reactive. An agile system's response to a change in the environment, whether to take advantage of an opportunity or to respond to a threat, is achieved, metaphorically, by reshaping the system so that it is compatible or synergistic with the changed shape of the environment. A reactive response is a compulsory systemic counter to a threatening change in the environment, with purpose to maintain or restore competitive functional performance. A proactive response is a non-compulsory systemic initiation enabled by a change in the environment, with purpose to improve competitive functional performance.

Resilient and Composable are the engineered cornerstones of the other terms explored earlier: Adaptive, Agile, and Autonomic.

Agile System Architecture (Dove 2014) – We are all very familiar with architectures that accommodate and facilitate structural change. Think of the construction sets we grew up with: Erector/Meccano sets (Figure 3), Tinker Toy, and Lego, to name some of the classics. Each of these construction sets consists of different types of components, with constraints on how these components can be connected and interact. Though each construction set is better suited to some types of constructions than others are, they all share a common architectural pattern. This author credits a master's student for the insightful comment: "It took a couple of days, but it finally occurred to me what agility was all about with regard to systems engineering. I realized that when your developing an agile system, you're really developing the "erector set" that can be assembled into a Farris wheel, as opposed to just building a Farris wheel."

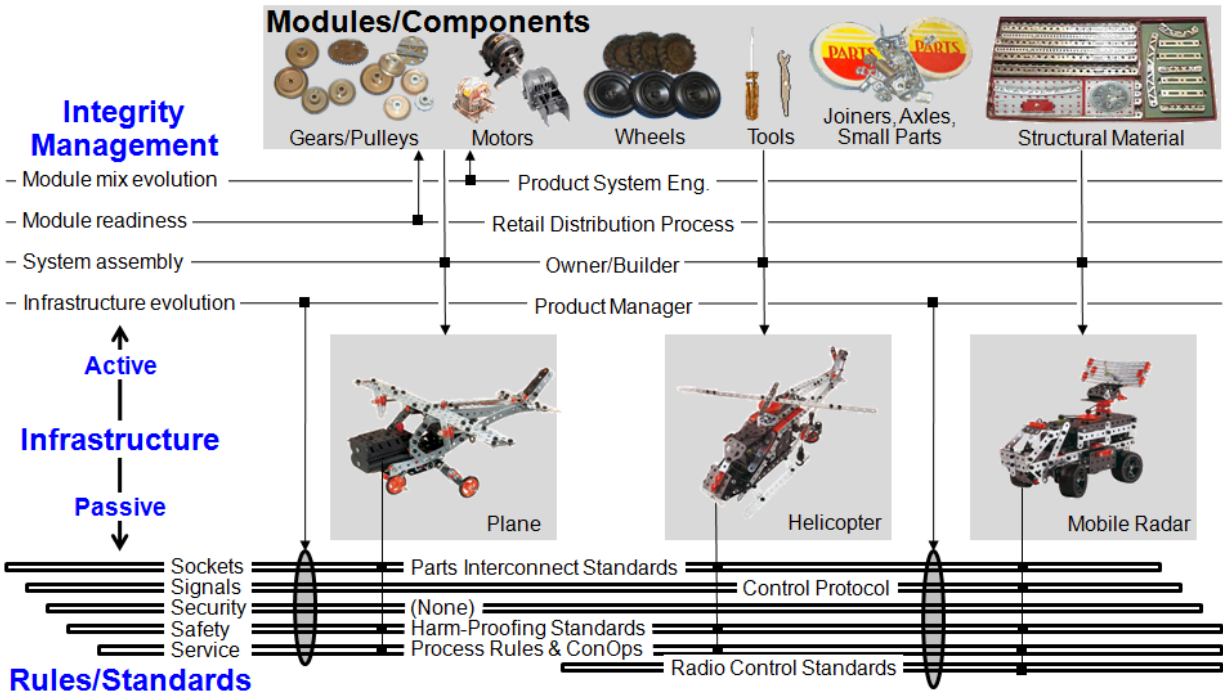

Figure 3. Agile architecture pattern (AAP) as manifested by Erector/Meccano sets.

There are three critical elements in the agile architectural pattern: an inventory of drag-and-drop encapsulated modules, a passive infrastructure of minimal but sufficient rules and standards that enable and constrain plug-and-play interconnection, and an active infrastructure that designates four specific responsibilities for sustaining agile operational capability.

Here the word module is a generic term for a human or inanimate drag-and-drop system asset.

- Modules—Modules are self-contained encapsulated units complete with well-defined interfaces, which conform to the plug-and-play passive infrastructure.
- Passive Infrastructure—The passive infrastructure provides drag-and-drop connectivity between modules. Its value is in isolating the encapsulated modules from each other so that we minimize unexpected side effects when changing modules and new operational functionality is rapid. We should consider at least five categories of standards and rules: sockets (physical interconnect), signals (data interconnect), security (trust interconnect), safety (of user, system, and environment), and service (Concept of Operations for system assembly and evolutionary agility sustainment).
- Active Infrastructure—An agile system cannot be designed and deployed in a fixed event and then left alone. Agility is most active as new system configurations are assembled during operation in response to new requirements – something that may happen very frequently. In order for new configurations to be enabled when needed, four actively dispatched responsibilities are required:
  - Module Mix Evolution—Who (or what process) is responsible for ensuring existing modules receive upgrades, addition of new modules, and removal of inadequate modules, in time to satisfy response needs?
  - Module Readiness—Who (or what process) is responsible for ensuring that sufficient modules are ready for deployment at unpredictable times?
  - System Assembly—Who (or what process) assembles new system configurations when new situations require something different in functionality?
  - Infrastructure Evolution—Who (or what process) is responsible for evolving the passive and active infrastructures as new rules and standards become appropriate to enable next generation capability.

Inherent in these four active responsibilities has been the understanding that sensemaking, which triggers change decisions for evolution and assembly, is a distributed responsibility, with specific natures and assignments allotted to appropriate active-infrastructure parties. These sensemaking responsibilities are documented in the passive infrastructure Service element ConOps, and in the Security element requirements. Sensemaking is critical, but it is not a separate system-architecture responsibility, but is rather distributed among at least three of the active-infrastructure responsibilities as vigilance and decision making, which triggers the exercise of the those architecture-indicated responsibilities. The responsibility name, e.g., module mix evolution, designates a comprehensive “what”, with the “how” specified in at least two of the passive infrastructure elements. In contrast, (Alberts 2006, Figure 7, p63), in a command-control-sensemaking-execution conceptual-model process architecture, rightfully give first-class prominence to Sensemaking.

Agile System Design Principles – Ten common Reusable-Reconfigurable-Scalable (RRS) design principles were discovered in workshop analysis of existing agile systems in the nineties, and are now used to guide agile-system design strategy

(Dove 2014). These principles are split into three categories, with the understanding that a principle in one category often provides benefit in the other categories. Need and intent are briefly outlined for each principle, with the “intent” providing a general strategy for meeting the need, and the understanding that an augmented or related approach may be a better fit to a specific-system need. Entire papers could be written on the variations and nuances of each of these principles. It is left to a designer’s creative insight to adapt the essence of the principle to the system of interest.

Reusable Principles:

- Encapsulated Modules (Modularity)—Need: System assemblers want effective module replacement and internal change without side effects. Intent: Modules physically encompass a complete capability, and have no dependencies on how other modules deliver their capabilities.
- Facilitated Interfacing (Plug Compatibility)—Need: System assemblers want effective interfacing that facilitates integration and replacement of modules. Intent: Modules share minimal interface standards, and are readily inserted and removed.
- Facilitated Reuse—Need: System assemblers want effective module selection and acquisition that facilitates reuse. Intent: Available modules are identified by capability and requirements, and can be readily discovered and acquired for deployment.

Reconfigurable Principles:

- Peer-Peer Interaction—Need: System assemblers want effective communication among modules. Intent: Modules communicate directly on a peer-to-peer basis to avoid intermediary relay failure, content filtering, and time delay.
- Distributed Control and Information—Need: System assemblers want effective information-based operational decisions. Intent: Decisions are made where maximal situational knowledge exists, and relevant information is maintained local to decision making modules while accessible globally.

- Deferred Commitment—Need: System assemblers want to maintain effective response ability. Intent: Conserve the commitment and consumption of limited resources to the last responsible moment, in anticipation of future unpredictable events and uncertain response needs.
- Self-Organization—Need: Systems assemblers want effective adaptation of interacting modules. Intent: Module relationships are self-determined where possible, and module interactions are self-adjusting or self-negotiated.

Scalable Principles

- Evolving Infrastructure—Need: System assemblers want effective acquisition and deployment of new module capabilities. Intent: Passive infrastructure standards and rules are monitored for current relevance, and evolve to accommodate new and beneficial module types in anticipation of need.
- Redundancy and Diversity—Need: System assemblers want effective resilience under quantitative (need more of something) and qualitative (need something different) situational variance. Intent: Duplicate or replicable modules provides quantitative capacity options and fault tolerance options; diversity among similar modules provides situational fit options.
- Elastic Capacity—Need: System assemblers want to incrementally match committed system resources to situational capacity needs of unpredictable or uncertain range. Intent: Modules may be combined in unbounded quantities, where possible, to increase or decrease deliverable functional capacity within the current architecture.

Agile Natural Systems – Offering Patterns for Agile Security

A project of INCOSE's Systems Security Engineering working group has been developing a collection of natural-system security patterns (Dove 2010a, 2010b), principally with master's students from Stevens Institute of Technology completing course term projects and Master's project.

The project is motivated by the observation that the adversarial community is leading the generally-reactive security community in both attack innovation and rapid evolution; accomplishing this with effective self-organizing behaviors. Six characteristics (Table 1) that underpin adversarial evolutionary and innovative behavior have been abstracted to for qualifying candidate self-organizing security patterns that may level the playing field.

Rather than review the 15 patterns already developed, see (Dove 2012b) for a published reference list at the end. One example, not yet a part of the project's pattern collection. will be discussed.

Natural organisms are composed of billions of cells. An organism has many identical and many different types of cells. Nature works with expendable resources, both at the cellular level and at the species level. Evolution occurs with experimentation and natural selection, enabling replication of those organisms that successfully compete in their environment and extinguishing those that don't. There is no purpose here. Nature is indifferent. What we observe is simply the ubiquitous algorithm of natural selection (Dennett 1995).

One autonomic approach to unnatural (engineered) systems leveraging this cellular model might compose a network with a sufficient number of same type network nodes; and employ a mechanism that seeks and identifies malfunctioning nodes, removes them from the system, and replaces them with fresh newly minted replicates. In the natural organism functional redundancy is high (an organ is composed of many identical cells working collectively), and replacing ineffective or bad cells is generally a constant process that doesn't impair organ functionality. Natural selection arrived at this construct through eons of experimentation.

Another example leveraging a natural system cellular model is offered by the brain's cortex, a general purpose neural network that allocates greater or lesser numbers of neurons to specialized brain functions (e.g., visual cortex, auditory cortex, sensemaking pattern recognition, expertise) depending upon usage frequency and time. Your total cell count (and subsequent weight) also varies as muscle, organ, and fat cells wax and wane in count for whatever cell-generating reasons come into play.

To make this cellular model work, a system functional architecture would depend on cellular redundancy that collectively achieves the intended function even while some or many cells are ineffective. This is not to be confused with functionally redundancy, but is rather redundancy in the achievement of a function. Network nodes could be repurposed to do anything, with the quantity of nodes devoted to similar functions varying over time, limited only by the total number of nodes available, which could also be expanded and contracted. Your muscles get larger with employment, and atrophy with little use. Cloud computing already offers this general model.

Natural systems employ the same fundamental agile system architecture previously outlined, enabling reliable highly complex and evolving system capability and functionality (Doyle 2004, 2010).

Agile Security for C2, Cyberphysical Systems, and Enterprise Support Systems

Command and control of cyberphysical systems is attracting the attack community. The now infamous Stuxnet attack ushered in a new era of attack sophistication and cyberphysical system targeting, going after the centrifuges with a damage intent, not the information system with a stealing intent. Stuxnet provided the call for cyberphysical C2 concern, and the more recent German steel-factory attack provided highly visible confirmation (Zetter 2015). Attacking command and control systems is generally motivated to cause disruption and damage, rather than

economic gain. Information stealing can hurt economically, but an attack on C2 can reach out and touch somebody physically, whether device targeted or information flow targeted.

Figure 3 shows a notional application of the fundamental Agile Architecture Pattern to an agile security domain, regardless of whether that domain is military C2, industrial C2, or an enterprise information support system.

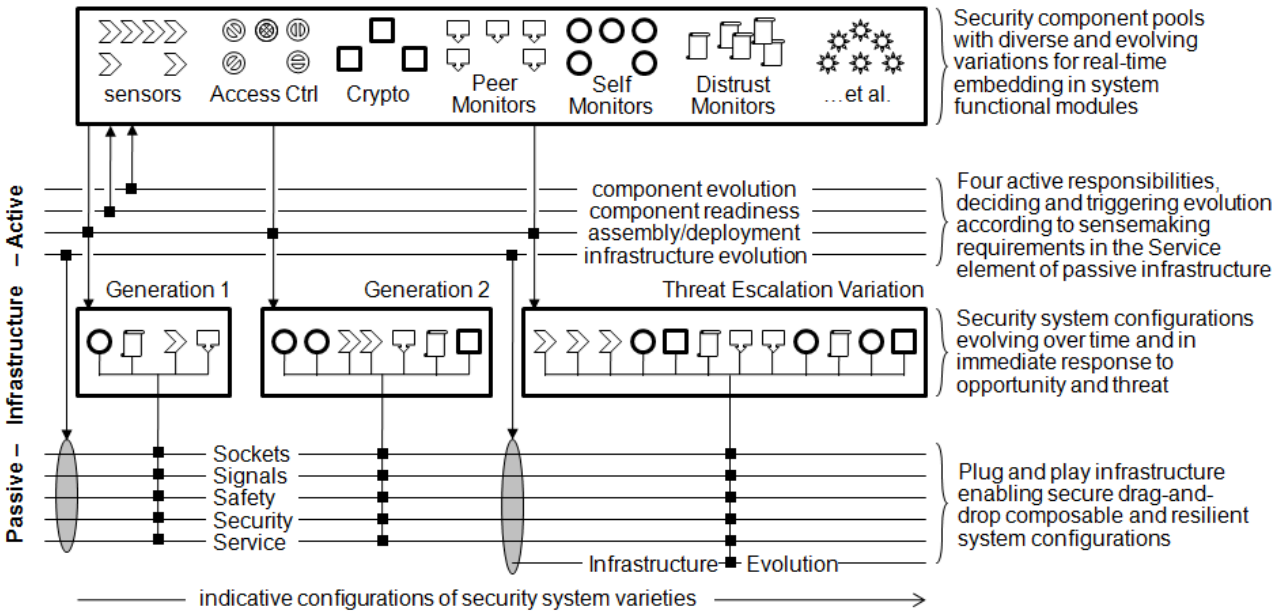

Figure 3. Notional agile architecture pattern for security reconfiguration/augmentation/evolution

Figure 4 shows notional encapsulated examples, assembled from the components of Figure 3 to protect not only the system perimeter, but also the individual functional components and sub-systems, a need outlined in the next section.

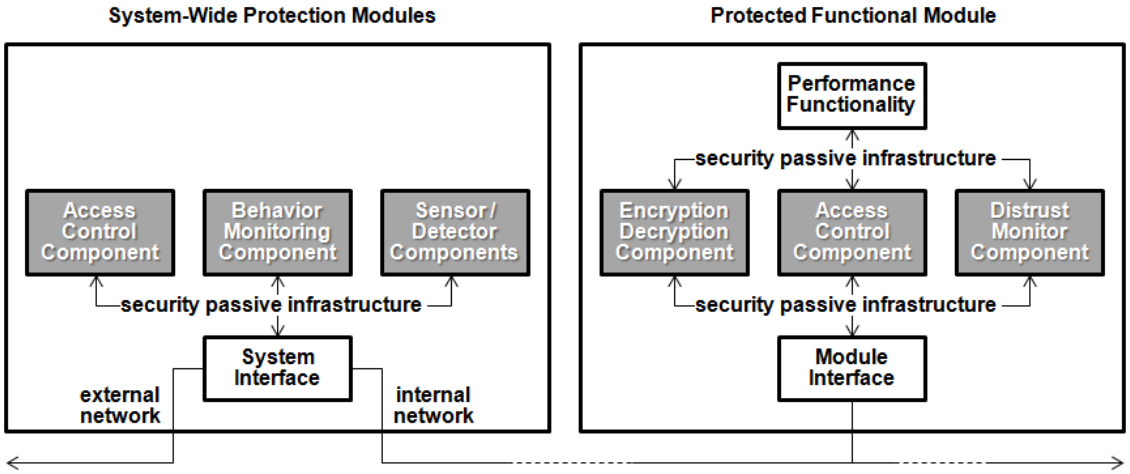

Figure 4. Notional concept – common security infrastructure enables rapid composable and resilient reconfiguration/augmentation/evolution.

Agile Security Considerations

This section suggests some key considerations for next generation security, characterized principally as a shift in emphasis from static to dynamic security, from after the act reaction to before the act prevention. In the current calls for security evolution, little to nothing is said of harmony and behavior monitoring, and too much is expected of trust, and human compliance. In current planning, these appear to be shortfalls: a failure to get what is expected or needed.

The author feels the considerations listed below are necessary for next-generation agile security, but not yet sufficient. The intension below is to express these considerations in minimal but concise words, leaving large latitude on how the intent might be achieved in satisfaction of the need:

- Holistic Systems Engineering – Need: Effective security reconfiguration/augmentation/evolution. Intent: Holistic systems thinking, security embedded in system design as part of system functionality, security system architecture structured and designed for adaptability and evolution.
- Collective Intelligence – Need: Shared knowledge base. Intent: Knowledge and experience shared and assimilated individually for organic thought and action. See (Boyd 1987).
- Harmony – Need: Embraceable rather than enforceable security. Intent: Support rather than inhibit human and organizational productivity and goal achievement.
- Self-Organization – Need: Response capability at cyber-speed. Intent: Response decision and action automated or human-enabled at the point of sufficient knowledge.
- Consistency – Need: Eliminate undependable security functionality. Intent: systemically automated security devoid of reliance on human compliance.
- Distrust – Need: Safely employ people and components that change asynchronously over time. Intent: Component-level self protection that distrusts all interaction. See (Kott 2014) for acknowledgement of the need for mixed trust systems.
- Shape Shifting – Need: Static system architecture can be observed and probed over time to discover vulnerabilities. Intent: Moving target defense (change functional methods) and offense (evolve capability and functional methods). See (Horowitz 2012:23-25) for defense.
- Component Conscience – Need: Component self awareness and evaluation of behavior. Intent: Self monitoring internal conscience as an embedded independent agent. See (Dove 2009a, 2009b, 2012a; Horowitz 2015:73-105).
- System Conscience: Need: Emergent system and system-of-systems behavior can occur unpredictably. Intent: System-wide emergent behavior monitoring by independent agents.
- Peer Behavior Judgment – Need: Aberrant component operational behavior can be caused by design flaws, system malfunction, human error, and malevolent control penetration. Intent: Peer-peer behavior monitoring. See (Dove 2009a, 2009b, 2012a).

Conclusions (This section not completed)

Agile security is designed for change. It can be augmented with new functional capability. It can be restructured with different internal relationships among subsystems. It can be scaled up or down for economic delivery of functional capability. Agile security can be reshaped to regain compatibility or synergy with an environment that has changed shape, or an environment that offers innovative opportunity. These types of changes are structural in nature, and require an architecture that accommodates structural change. A system engineer tasked to design an agile system in some functional domain starts with the design of the erector set architecture for that domain.

As can be seen by the review of terms section, agile security is in a vision-setting and early development stage, but not smoke and mirrors, as funding sources and considerable effort are being applied to realize these necessary capabilities. Agile security encompasses a family of related initiatives. To be effective they will all share the equivalent of common DNA – a fundamental agile architecture pattern with fundamental design principles. This is their common family root, and needs attention at the infrastructural definition area to strengthen the family and each of its members.

One way to converge on an appropriate and common family root was proposed in 2005 by the Agile Security Forum (Dove 2004, 2005a, 2005b). That proposal sought convergence in the industrial rather than the government domain, and was ahead of its time. Nevertheless, there is food for timely thought there.

In closing, to air a peeve about terminology, this discussion has been about *cyber security*, not *cyber*, as used too often by some in the security community. The Oxford English Dictionary defines cyber as an adjective, not a noun – “Of, relating to, or characteristic of the culture of computers, information technology, and virtual reality. Origen 1980s: abbreviation of cybernetics, a plural noun treated as a singular – The science of communications and automatic control systems in both machines and living things.”

References (all links accessed 5-Feb-2015)

AFRL. 2014. BAA-Rik-14-12 2014: Command and Control of Proactive Defense (C2PD). Department of the Air Force, Air Force Materiel Command. www.fbo.gov/spg/USAF/AFMC/AFRLRRS/BAA-RIK-14-12/listing.html

Alberts, David S. 1996 revised 2002. Information Age Transformation – Getting to a 21st Century Military. DoD Command and Control Research Program (CCRP). www.dodccrp.org/html4/books_downloads.html

Alberts, Davis S., Richard E. Hayes. 2006. Understanding Command and Control. DoD Command and Control Research Program (CCRP). www.dodccrp.org/html4/books_downloads.html

Alberts, David S. 2011. The Agility Advantage: A Survival Guide for Complex Enterprises and Endeavors. DoD Command and Control Research Program (CCRP). www.dodccrp.org/html4/books_downloads.html

Bodeau, Deborah J., Richard Graubart. 2011. *Cyber Resiliency Engineering Framework*. MITRE Technical Report: Document Number MTR110237. MITRE Corporation Bedford, MA. September. www.mitre.org/sites/default/files/pdf/11_4436.pdf

Boyd, John, 1987. Organic design for command and control. One of the briefings in *A discourse on winning and losing*. Maxwell Air Force Base, AL: Air University Library Document No. M-U 43947. http://dnipogo.org/john-r-boyd

Collier, Zachary A. Igor Linkov, Daniel DiMase, Steve Walters, Mark (Mohammad) Tehranipoor, James H. Lambert. 2014. Cybersecurity Standards: Managing Risk and Creating Resilience. *Computer*, IEEE Computer Society, September.

Cebrowski, Arthur K. 2003. Military Transformation: A Strategic Approach. U.S. Department of Defense, Office of Force Transformation. www.iwar.org.uk/rma/resources/transformation/military-transformation-a-strategic-approach.pdf

Dennett, Daniel C. 1995. *Darwin's Dangerous Idea – Evolution and the Meanings of Life*. Simon & Schuster.

DHS. 2014. Enterprise Automated Security Environment (EASE) Request for Information (RFI). Solicitation Number: RFI201411, Department of Homeland Security. December 3. www.fbo.gov/index?s=opportunity&mode=form&id=7497164c010cce00d2f1ce6db79c6727&tab=core&_cview=1

DoD. 2011. Department of Defense Strategy for Operations in Cyberspace. U.S. Department of Defense. July 2011.

Dove, Rick and Roger Nagel (Principle Investigators). 1991. 21st Century Manufacturing Enterprise Strategy – An Industry-Led View (Volume 1) and – Infrastructure (Volume 2). Eds: S. Goldman and K. Preiss. Diane Publishing Company. www.parshift.com/s/21stCenturyManufacturingEnterpriseStrategy-Volume1.pdf, www.parshift.com/s/21stCenturyManufacturingEnterpriseStrategy-Volume2.pdf.

Dove, Rick. 1992. The 21st Century Manufacturing Enterprise Strategy or What is All This Talk about Agility? Invited paper originally published by Paradigm Shift International (December) and then translated into Japanese and published in a 1993 issue of Prevision, the Japan Management Association Research Institute.

Dove, Rick. 1993. *Beginning the Agile Journey – A Guidebook*. Hewlett Packard. www.parshift.com/Files/PsiDocs/Pap930701Dove-BeginningTheAgileJourney-A Hewlett Packard Guidebook.pdf

Dove. Rick. 2001. Response Ability – The Language, Structure, and Culture of the Agile Enterprise. Wiley.

Dove, Rick, 2004. Rectifying the Information Security Gap. Agile Security Forum. November 10. www.agilesecurityforum.com/docs/AsfPaperConceptCall.pdf

Dove, Rick. 2005a. Frameworks for Analyzing and Developing Agile Security Strategies - Oriented for the Energy and Utility Sector. Agile Security Forum. January 22. www.agilesecurityforum.com/docs/AsfPaperSixFrameworks.pdf

Dove, Rick. 2005b. Pathfinder Initiative – Concept of Operations. Agile Security Forum. January 26. www.agilesecurityforum.com/docs/AsfPaperConceptOps.pdf

Dove, Rick. 2009. Embedding Agile Security in System Architecture. *Insight* 12 (2): 14-17. International Council on Systems Engineering, July. www.parshift.com/Files/PsiDocs/Pap090701Incose-EmbeddingAgileSecurityInSystemArchitecture.pdf

Dove, Rick. 2009a. Paths for Peer Behavior Monitoring Among Unmanned Autonomous Systems. ITEA Journal 30: 401–408, September. www.parshift.com/Files/PsiDocs/Pap090901IteaJ-PathsForPeerBehaviorMonitoringAmongUAS.pdf

Dove, Rick. 2009b. Methods for Peer Behavior Monitoring Among Unmanned Autonomous Systems. ITEA Journal 30: 504–512, December. www.parshift.com/Files/PsiDocs/Pap091201IteaJ-MethodsForPeerBehaviorMonitoringAmongUas.pdf

Dove, Rick, Laura Shirey. 2010a. On Discovery and Display of Agile Security Patterns. 8th Conference on Systems Engineering Research. Hoboken, NJ, March 17-19. www.parshift.com/Files/PsiDocs/Pap100317Cser-OnDiscoveryAndDisplayOfAgileSecurityPatterns.pdf

Dove, Rick. 2010b. Pattern Qualifications and Examples of Next-Generation Agile System-Security Strategies. IEEE International Carnahan Conference on Security Technology (ICCST), San Jose, CA, USA, 5-8 Oct. www.parshift.com/Files/PsiDocs/PatternQualificationsForAgileSecurity.pdf

Dove, Rick. 2012a. Righteousness and Conscience as a Path to Socially Acceptable Autonomous Behavior. Insight 15 (2): 32-34. International Council on Systems Engineering, July. www.parshift.com/s/120701Insight-RighteousnessAndConscience.pdf

Dove, Rick. 2012b. Introduction to Self-Organizing Adaptive Systems. Plenary presentation at International Carnahan Conference on Security Technology. Boston, MA. October 15-18. www.parshift.com/s/121015ICCST-Patterns.pdf

Dove, Rick, 2013. Sustainable Agile Security Enabled by Systems Engineering Architecture. *Insight* 16 (2): 30-33. International Council on Systems Engineering, July. www.parshift.com/s/130701Insight-EnablingSustainableAgileSecurity.pdf

Dove, Rick and Ralph LaBarge. 2014. Fundamentals of Agile Systems Engineering – Part 1. International Council on Systems Engineering IS14, Los Vegas, NV, 30-Jun-03Jul. www.parshift.com/s/140630IS14-AgileSystemsEngineering-Part1.pdf

Dove, Rick. 2015. INCOSE Corporate Advisory Board – Agile SE Break Out Status and Plans. Unpublished presentation. January 24. www.parshift.com/s/150124IW-CABAgileSEBreakOutStatusAndPlans.pdf

Doyle, John. 2004. Protocols. In Willinger, W. & Doyle, J. C. *Robustness and the Internet: Design and evolution. in Robust Design: A Repertoire of Biological, Ecological, and Engineering Case Studies*, ed. Jen, E. (Oxford Univ. Press, New York), 231-272.

Doyle, John C. 2010. The Architecture of Robust, Evolvable Networks – Organization, Layering, Protocols, Optimization, and Control. The Lee Center. Spring. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.400.6626&rep=rep1&type=pdf

Emami-Tabla, M., M. Amoui, L. Tahvildari. 2013. On the Road to Holistic Decision Making in Adaptive Security. *Technology Innovation Management Review*. August 2013: 59–64.

Fowler, Martin and Jim Highsmith. 2001. The Agile Manifesto. Dr. Dobb's Journal, August. www.drdobbs.com/open-source/the-agile-manifesto/184414755.

Gertz, Bill. 2013. NSA Document: Cyber attacks are asymmetric weapons and significant future threat to U.S. November 23. http://flashcritic.com/nsa-cyber-attacks-are-an-asymmetric-warfare-weapon-and-pose-a-significant-threat-facing-the-u-s/

Herring, Michael; Keith D. Willett. 2014. Active Cyber Defense: A Vision for Real-Time Cyber Defense. *Journal of Information Warfare*, 13(2):46-55, April. www.nsa.gov/ia/_files/JIW-13-2--23-April-2014--Final-Version.pdf

Horowitz, Barry (PI). 2012. Security Engineering Project. A013 - Final Technical Report SERC-2012-TR-028-2, 23-25. October 24. www.dtic.mil/dtic/tr/fulltext/u2/a582703.pdf

Horowitz, Barry (PI). 2015. System Aware Cyber Security for an Autonomous Surveillance System On Board an Unmanned Arial Vehicle. Systems Engineering Research Center. Technical Report SERC-2015-TR-036-4 Part 1a, 73-105. January 31. http://serc2cdn1.mannadesignworks.netdna-cdn.com/wp-content/uploads/2014/11/SERC-RT-115-Security-Engineering-Pilot-Final-Report-SERC-2013-TR-036-4-Parts-1a-1b-3-4-20150131.pdf

IBM. 2001 Autonomic Computing: IBM's Perspective on the State of Information Technology. IBM. http://people.scs.carleton.ca/~soma/biosec/readings/autonomic_computing.pdf

INCOSE. 2014. *A World in Motion – Systems Engineering Vision 2025*. International Council on Systems Engineering. June. www.incose.org/ProductsPubs/products/sevision2025.aspx

ISO/IEC/IEEE. 2008. Systems and software engineering — System life cycle processes, Second edition 2008-02-01, Software & Systems Engineering Standards Committee of the IEEE Computer Society.

Kadtke, James, Linton Wells II. 2014. DTP 106: Policy Challenges of Accelerating Technological Change: Security Policy and Strategy Implications of Parallel Scientific Revolutions. Center for Technology and National Security Policy, and National Defense University. September 12.

Kephart, Jeffrey O., David M.Chess. 2003. The Vision of Autonomic Computing. Computer, IEEE Computer Society. January.

Konda, Matthew. n.d. Security for Software Developers. Jemuri. http://jemurai.com/agile-security.html

Kott, Alexander, Ananthram Swami, Patrick McDaniel. 2014. Security Outlook: Six Cyber Game Changers for the Next 15 Years. IEEE Computer Society, Computer, 47(12):104-106, December.

Lawson, Harold 'Bud'. 2010. *A Journey Through the Systems Landscape*. College Publications.

Musman, Scott, Seli Agbolosu-Amison. 2014. A Measurable Definition of Resiliency Using "Mission Risk" as a Metric. MITRE. February.

Savage, Julie DelVecchio. 2010. Composable Capability on Demand (CCOD®): A New Paradigm for the Design, Acquisition and Employment of IT-Based C2. 15th International Command and Control Research and Technology Symposium, Santa Monica, California, June 22-24. www.dodccrp.org/events/15th_iccrts_2010/papers/102.pdf accessed 2Feb2015.

Sourcefire. n.d. www.sourcefire.com/agile Site is unavailable (will track it down later and change or substitute).

Willett, Keith. 2015a. Integrated Adaptive Cyberspace Defense: Secure Orchestration. Pending publication: 20th International Command and Control Research and Technology Symposium, Annapolis, MD, June 16-19.

Willett, Keith D. 2015b. Adaptive Knowledge Encoding for Agile Cybersecurity Operations. Submitted: International Council on Systems Engineering International Symposium, Seattle, WA, July 13-16.

Zetter, Kim. 2015. A Cyberattack Has Caused Confirmed Physical Damage for the Second Time Ever. January 8. Wired Magazine. www.wired.com/2015/01/german-steel-mill-hack-destruction