Reference Framework and Model for Integration of Risk Management in Agile Systems Engineering Lifecycle of the Defense Acquisition Management Framework

Portia Crowe
732-427-5757
pcrowe@stevens.edu

Ali Mostashari, PhD
201-216-8723
amostash@stevens.edu

Mo Mansouri, PhD
201-216-8719
mo.mansouri@stevens.edu

Robert Cloutier, PhD
201-216-5378
robert.cloutier@stevens.edu

Castle Point on Hudson, Stevens Institute of Technology, Hoboken, NJ USA 07030

Abstract. Risk management is a requirement for success of any enterprise. Despite this fact, there have been limited solutions for integrating risk management in agile life cycles. On average, 70% of all IT-related projects fail to meet their objectives which have caused 80% of companies to reassess their process. One of these processes includes risk management where studies have shown that risks should be assessed in all dimensions of a program. We found that risk management was being conducted on the overall program with concentration in the software development phase which leaves other life cycle phases with no risk assessments and mitigation strategies. Agile systems engineering is more complex than traditional systems engineering due to events happening concurrently and non-sequentially during the life cycle of a program. This paper will propose a framework and model for risk 'managility', management in an agile environment, specifically in systems engineering life cycle. We use the Department of Defense's Evolutionary Acquisition and Management Framework as a bridge to define and illustrate an agile systems engineering life cycle (ASEL) framework and an agile risk integrated engineering lifecycle (ARIEL) model.

1. Introduction

On average, 70% of all IT-related projects fail to meet their objectives which have caused 80% of companies to reassess their process (Verman 2007). Conducting risk management throughout a program's life cycle can prevent failed projects by mitigating risks and tracking them. Agile cyclical models claim to be risk-driven because of the fact that risks determine the path development or a program must take to rapidly field capabilities (Boehm and Turner 2004). Thus, risk management should be used to define and address risks while using agile methods. The popularity of agile development has brought about changes in systems engineering processes and the dynamics of evolutionary management throughout the life cycle. Exposing risks within an agile life cycle is critical due to the rapid nature of fielding capabilities and the expected outcome of performance. Systems are getting more complex as is the need to build, manage, and support those systems. Complexity is multiplied when dispersing new technology into the world while having to train and support new capabilities. We propose to show how risk management can evolve from traditional static implementation to an agile and flexible integrated motion continuously throughout a program's life cycle.

Risks can manifest quickly so tracking within each phase of a program's life cycle is essential.

However, what we found is that most risk management activities are heavily placed on the overall program and during the software development phase leaving other phases exposed to risk without mitigations. Risk management can be detailed and have prolonged processes whereas agility relies on rapidness. Although The Department of Defense's Evolutionary Acquisition (DoD EA) approach modernizes the traditional life cycle approach, the risk management framework has not changed and conforms to the traditional implementation methods thus leaving a void that is critical to invoking success.

We investigated how risk management should be integrated into an agile systems engineering lifecycle (ASEL). To do this, we researched well-known risk management standards to show the void in relating the standards to agile systems engineering life cycle. We propose a reference framework and model that depict integration of risk management into an ASEL. The agile life cycle model shows integration of DoD life cycle phases with risk management to enable continuous user and stakeholder involvement and rapid fielding of capabilities.

We provide a comprehensive life cycle framework using The Defense Acquisition Management Framework life cycle phases as our basis. We used the DoD EA Approach as our basis for the development to show a more specific and less ambiguous model for delivering capabilities incrementally. The model will depict integration of risk management for each phase and within these phases. We propose a simplistic grouping of systems engineering phases which are:

- Phase 1: Concept Refinement, Requirements & Architecture Analysis, Design
- Phase 2: Capability Development, Integration & Test, Demonstration
- Phase 3: Production & Deployment, Operations, Support & Training

2. Background and Literature Review

A. Risk Management

To propose integration of risk management in agile systems engineering life cycles, we must understand what risk management is. Risk is the possibility of loss or injury (Merriam-Webster 2008). Organizations that track risks traditionally outperform those who do not. Managing risks is more vital than ever where timing, reliability and cost effectiveness are key measurements of success. According to Lt. Gen. Harry D. Raduege Jr.,USAF (Ret.):

It is a rare organization that intelligently manages the full spectrum of risks; that adequately assesses and addresses risk from all angles; that breaks through organizational barriers that prevent an objective assessment; and that systematically anticipates and prepares a holistic plan to manage potentially significant risks (Raduege 2008).

There are many interpretive models that depict risk management processes and they often include risk identification, assessing, handling, and monitoring (see figure 1) (US DoD 2006). In agile life cycles, risk is predominant due to the nature of agility across many phases working in parallel and shortened cycles. This inherently cause risk management to be a critical process to ensure success.

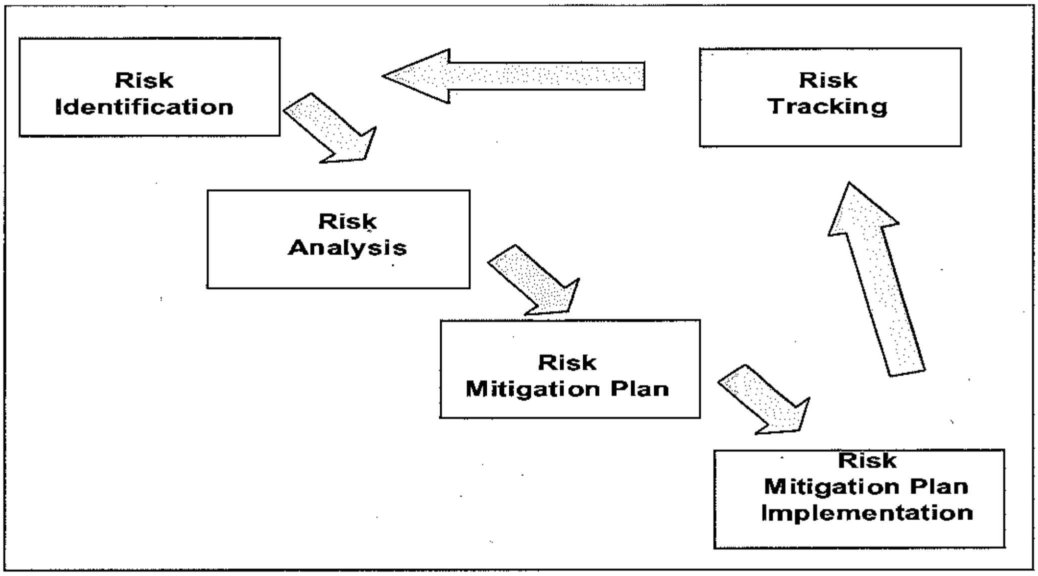

Figure 1. DoD Risk Management Process (US DoD 2006)

Through literature reviews, we have found that risk management predominantly occurs for the overall program and in the software development phase. The IEEE 1540 Risk Management process model and requirements are complemented by other standards such as IEEE 12207 Software Life cycle (IEEE 2001, IEEE 1998). However, the two standards together do not depict an entire life cycle but mainly focus on software development, nor does it account for agility in the life cycle model. Other risk management organizations have described standards and process for life cycles but have mainly focused on the process for the overall program or mainly on software development (Boehm 1991, Boehm et al 1988, Military 2006, Nyfjord and Kajko-Mattsson 2007, Sommerville 2006, Williams et al 1999). Risk management in system of systems has emphasized common issues which include multiple stakeholders, multiple process, long life cycles and integration risk at the system level (Boehm and Hansen 2001, Conrow 2005). The DoD's recently published systems engineering guide states the following example for system of systems risk management:

> *Risk management is a core function of SE at all levels; consequently, it appears in all SoS SE elements. In Translating Capability Objectives, the systems engineer evaluates the specified capabilities and assesses the viability (and associated risk) of meeting SoS objectives, given the results of other SoS SE core elements (US DoD Systems 2008).*

At the enterprise level, risk management has been studied for collaborative process improvement and reengineering (Karduck et all 2007). In the field to date, risk management has been focused on software-intensive projects and on the software development phase. However, failure to address risk management during all phases of a program's life cycle, especially if it is agile, can mean failure to a program or incur additional costs (Boehm 1991, Boehm et al 1988, Conrow 2005, Hansen et al 2001).

B. Agile Systems Engineering Life Cycle with RM

What does agile systems engineering life cycle mean? Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems that integrates all the disciplines in a team effort forming a structured development process that proceeds from concept to production to operation (INCOSE 2004). Every program has a life cycle. In systems engineering that means an overall process from initial requirements through system and software development to maintenance of the product. An agile systems engineering life cycle is the rapid approach that integrates all phases of the product's life cycle. There are detailed recipes for balancing agility and discipline in systems and software engineering (Boehm and Turner 2004).

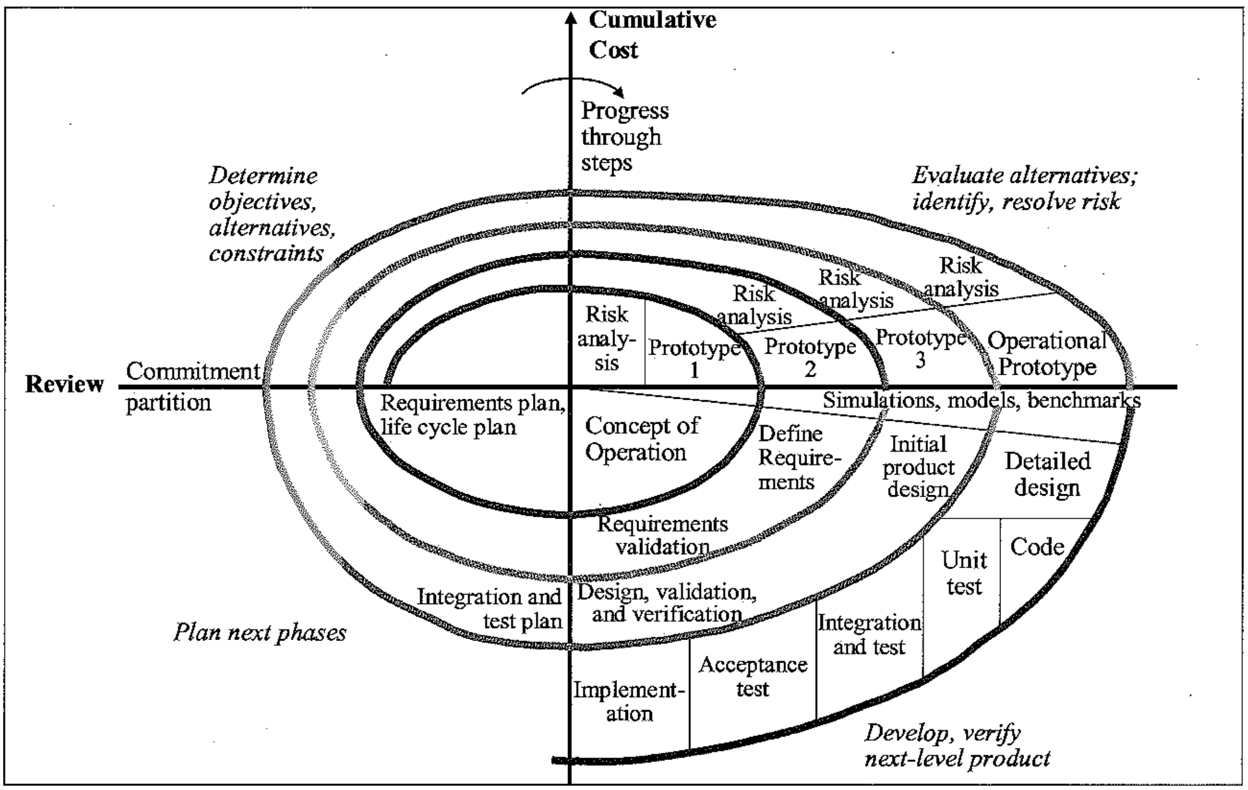

Figure 2. Win-Win Spiral Software Model (Boehm et al 1988).

In figure 2, Barry Boehm developed the spiral model to show how development and testing can be done while invoking risk analysis in every increment. Although, this model is focused on software development, the same can apply to all phases of a life cycle. Risk analysis is just one step of the risk management process but this diagram allows us to see the relationship of determining risks and reassessing it at every increment together with user feedback and demonstration to determine the best possible capability to be produced.

In today's complex environment, capabilities are needed much faster and at a lower cost than ever before. For example, The Global War on Terrorism (GWOT) has dictated that we are time-bound to respond to threats with new technology and economic capabilities that are simplistic vs. complex and that can be adaptive due to uncertainties and unforeseen requirements. One approach to balance imperative need with rapid response is an agile life cycle paradigm with integrated risk management tailored for cyclical and non-sequential acquisition to meet challenges brought about by changes in the environment and capabilities.

C. DoD Acquisition Framework and Process

The Defense Acquisition Management Framework process is a series of phases and decisions made for development of a system from material capability to a field/sustained system (US DoD 2007). In our research methodology for risk integrated ASEL, we evolve the DoD framework into a flexible framework to support agility throughout program lifecycles. When we reviewed the DoD framework, we found that all of the important systems engineering phases were captured and represented. Figure 3 illustrates the approved DoD framework to include all phases and milestones used for acquisition.

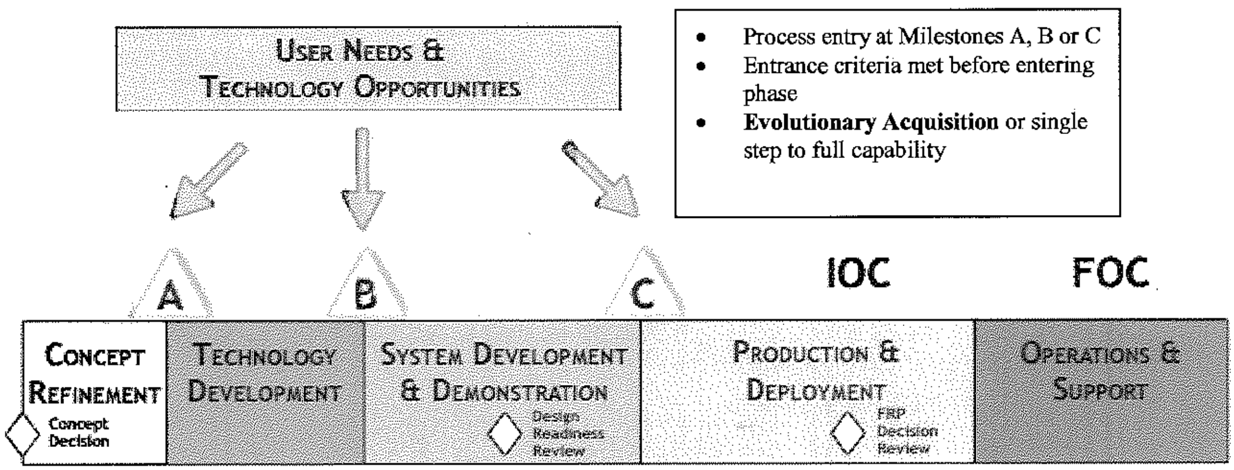

Figure 3. Defense Acquisition Management Framework (US DoD 2007).

The DoD management framework shows the phases of life cycle management from concept refinement to operations and support with milestone activities. The first 2 phases, concept refinement and technology development, are considered pre-system acquisition. System acquisition happens during phases 3 and 4, system development and demonstration and production and deployment. The final phase 5, operations and support, are considered sustainment (US DoD 2007).

The DoD EA approach is the preferred DoD strategy for acquisition by delivering capabilities in increments (US DoD 2003). However, the approach recommended is ambiguous which leads implementation to be diverse and debatable (Lorell et al 2006). In addition, detailed positioning of risk management within this acquisition strategy needs to be recommended if not imposed on the EA implementer, especially in incremental and spiral development where activities are risk driven. In the DoD Systems Engineering Guide, The Defense Acquisition Guide is the Risk Management Guide for DoD Acquisition all state that risk management is needed in the program's life cycle but mostly refer to managing risks in the overall program and within the software development phase (US DoD 2006, US DoD 2007, US DoD Systems Engineering 2008).

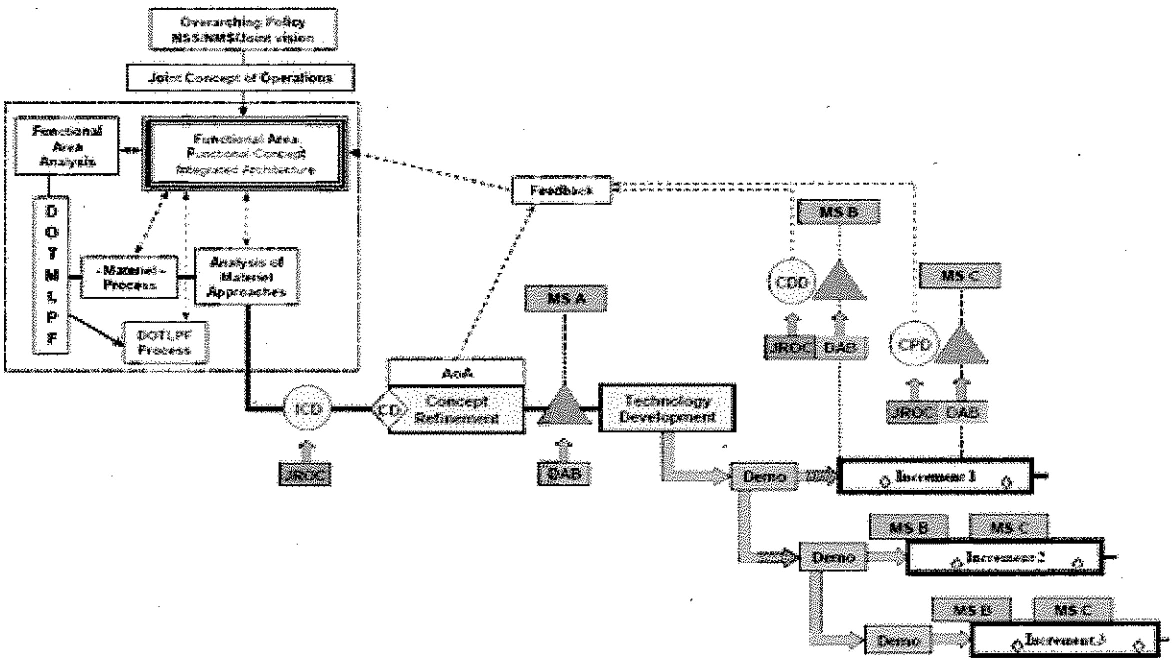

Figure 4. Requirements and Acquisition Process (US DoD 2003)

A shortcoming of the guidance regarding EA is that it lacks definition (Lorell et al 2006). Specific implementation approaches vary although it states that capability should be delivered in increments where the end state is known. Although DoD has taken a traditional life cycle approach and modernized it, the risk management framework has not changed and conforms to the traditional methods thus leaving a void that is critical to invoking agile balance and discipline.

3. Research Methodology

A. Framework

Agility is often found in different processes and methods. We propose an agile systems engineering life cycle as just one approach to rapid fielding. In our research, we found that there were key characteristics needed to bridge the DoD framework processes to agile framework processes to include the use of risk management and light weight processes. We have included the use of capability development instead of technology development. We have gathered our findings of key characteristics for utilizing an agile life cycle framework as follows:

Characteristics	Comments
Liberty to be dynamic	Agility needs dynamic processes while adhering to acquisition milestones
Non-linear; Cyclical and non-sequential	Life cycle behavior not like traditional waterfall models or linear frameworks; decreasing cycle times
Adaptive	Conform to changes such as capability and environment
Simultaneous development of phase components	Rapid fielding time may not lend to traditional phase containment (i.e. training and SW development together)
Ease of Change	Culture shift to support change neutrality; ease of modification built into architecture and design
Short Iterations	Prototyping, demonstrating and testing can be done in short iterative cycles with tight user feedback loop
Light-weight phase attributes	Heavy process reduction such as milestone reviews, demonstrations, and risk management
Capability Development to shape Technology Development	Capability needs determine requirements and technology used to achieve goals
Risk Management not just relating to software development/maintenance	Light-weight RM used within every phase and between them

Table 1. Bridging the DoD Framework to an Agile Life Cycle Framework

These developed characteristics bridge the void between the DoD framework and a flexible and agile life cycle framework. Changes in the complex world we live in have led us to evolve to a light-weight framework where phases are executed in parallel to save time and costs. However this will depend on working closely with the stakeholders of the program while imploring discipline throughout the cycle (Turner 2007).

The framework that we illustrate should not change the fundamentals of the DoD policy but support the DoD's evolutionary acquisition paradigm where capabilities are needed to be fielded faster in light of new threats and the changing environment we live in today. Capabilities can be built faster by allowing for flexibility and having a closer relationship with the customers and users of the system. Our proposed phases are loosely mapped to the DoD acquisition framework but goes further to show a non-linear and non-sequential relationship. We deduced that to stay in line with GWOT main initiatives, we wanted to incorporate capability development instead of technology development which we captured in our framework and model (Cherinka 2007). We propose the following framework as an approach to agile system engineering life cycle where phase components interact with each other instead of standalone activities to enforce agility:

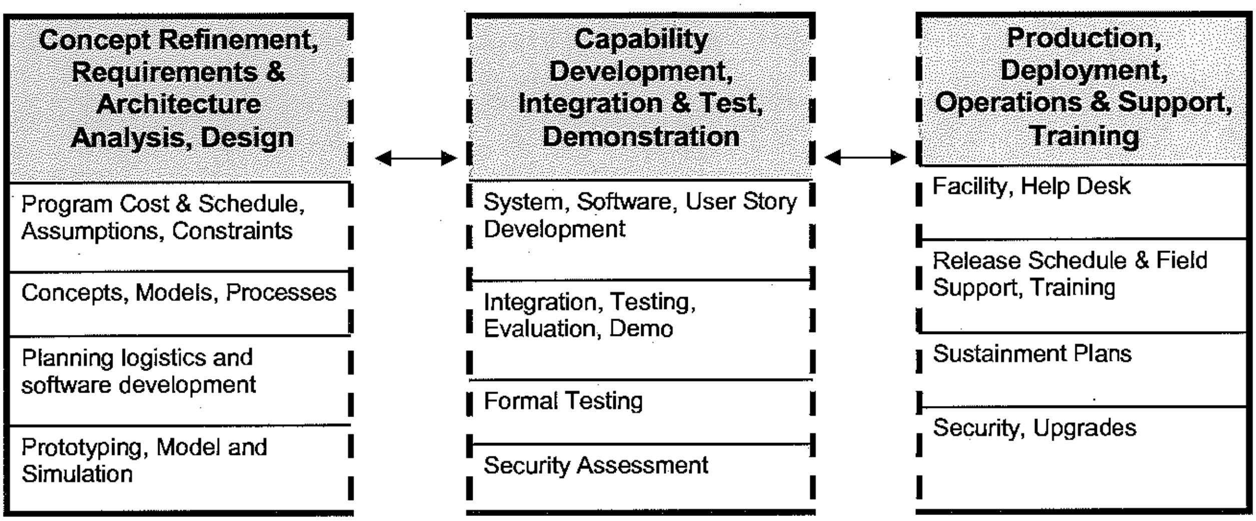

Figure 5. Agile Systems Engineering Life Cycle Framework

The key to the life cycle framework is to keep agility and flexibility in mind. We developed this framework using agile attributes such that all phases are somewhat 'rolling' into the other and while there are boundaries, they are not strict boundaries. The components are contained within each phase but not so much so that they are developed restrictively and linearly. Light-weight reduction and lean activities should support this framework to include reviews and risk management processes to facilitate agility and flexibility.

B. Model

In the DoD EA process, the concept refinement phase is bridged to the technology development stage through an acquisition milestone. Although we don't want to change the milestones that DoD has imposed on its acquisition approach, tailoring this approach to loop between these phases allows more agility and flexibility in formulating, extracting, and managing requirements and or capabilities. We also notice that each increment is treated like its own program where two milestones need be achieved before moving on to the next increment. This process is time consuming when schedules are aggressive, costly and suggests very heavy management processes where rapidness and flexibility are needed. Integration of risk management in this process appears to have the same affect. Each increment and milestone review imposed by the DoD EA introduces new risks. One of the key principles of the EA concept was to break down long and inflexible traditional acquisition programs into several much shorter and lower risk increments (Lorell et al 2006).

Interpreting the DoD EA strategy and process the best we can, we propose to define an implementation model that is characteristic of an agile systems engineering life cycle approach while integrating risk management throughout the model. Risk management is part of the process that must be conducted to balance the agility process with some normalization of penalties vs. value through out the life cycle. Together with our proposed ASEL framework, we propose an agile risk integrated engineering lifecycle (ARIEL) model that shows the integration of risk management throughout an agile lifecycle. We propose our ARIEL model as shown here in figure 6:

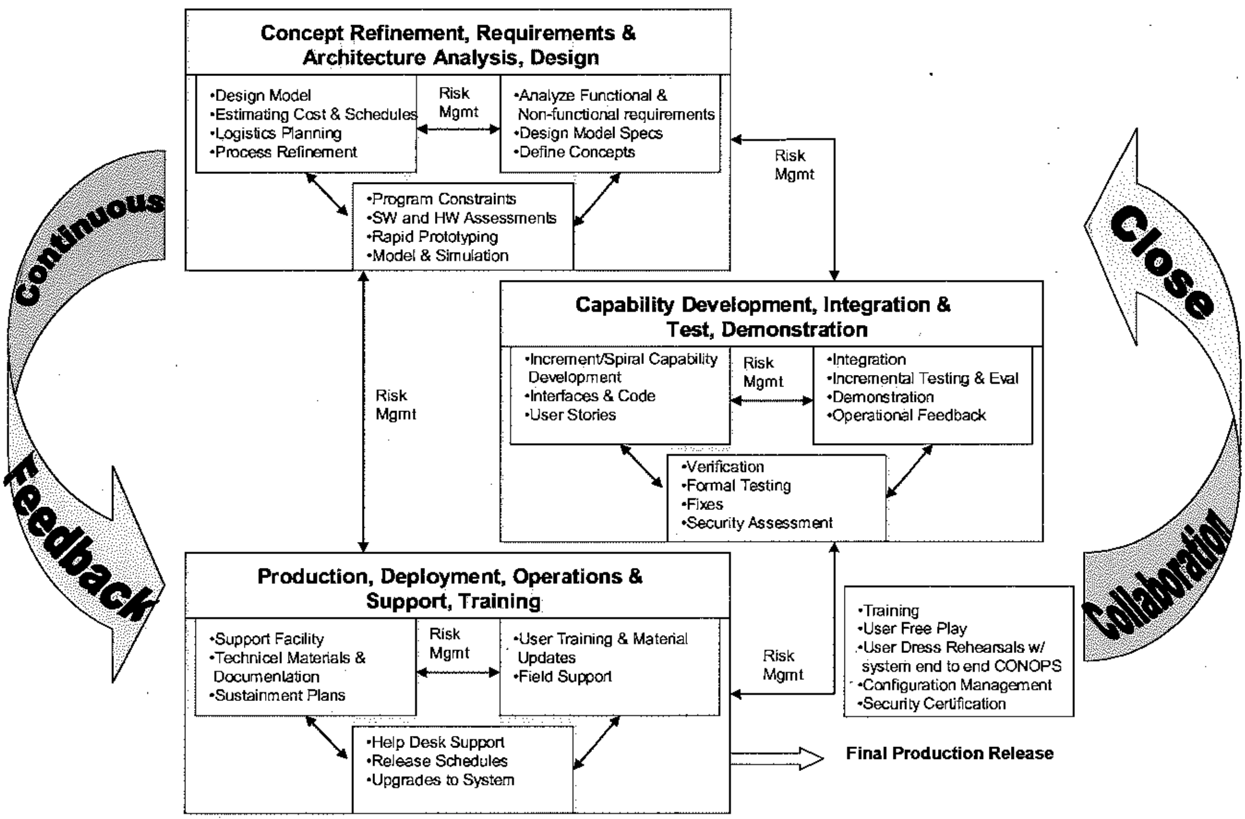

Figure 6. ARIEL (Agile Risk Integrated Engineering Lifecycle) Model

The ARIEL model shows a flexible and agile three-phase approach with close and continuous user and stakeholder feedback and integration of risk management. Phase 1 consists of concept refinement, requirements and architecture analysis and design. This phase should include estimating cost and schedule, logistics planning and program constraints. Phase 2 is the capability development, integration and test and demonstration phase where interface development and incremental testing and evaluation are being conducted along with security assessments. Phase 3 includes production and deployment and the operations and support, and training is proposed as phase 4. All phases include risk management which is often and cyclical to help reduce overall risk to the program.

In our agile systems approach, the life cycle functions allow concept refinement to continue while building capabilities and once stable enough, training and support evolve throughout the life cycle. The agile systems approach allows injection of new capabilities in a short amount of time while keeping user satisfied and well trained. Logistics is often the last thought about phase in the cycle. However, in agile systems engineering, we make it an essential part of the life cycle cyclical feedback loop which is risk managed and planned while the system evolves. In rapid fielding, the logistics has to be thought about in the beginning and during the development due to the time it takes to develop material and train the users adequately.

The continuous stakeholder and user feedback loop is an important part of ARIEL. This collaboration and feedback allows the program to gain invaluable insight into what works and what doesn't and also allows constant requirement checking to ensure the product is built with users needs as a key driver. Today's complex environment may cause unforeseen requirements which ARIEL can handle through the iterative and flexible processes of the built in loops. The underlying mechanism of ARIEL is the tight structure imposing balance and agility and the loosely coupled intercrossing of phases which allow rapid systems engineering processes to work together. The risk management built into ARIEL is a token that gets passed from phase to phase and within each phase and raises risk identification throughout the model.

4. Case Study

The US Department of Defense is facing challenges to rapidly deploy operational capabilities in complex environments while using an evolutionary acquisition construct. The US Army has developed capabilities that use an incremental approach and agile methodologies. We propose an illustrative case study using the Defense Readiness Reporting System-Army (DRRS-A) which is a secure web-based set of capabilities that include Army unit status reporting that details mission-critical information, providing Commanders with an accurate representation to make critical command and control decisions (Military 2006). The program fielded its first capability set in 2006 going from zero to 10 months using an agile approach. Continuous upgrades and additional capability sets are developed and fielded as quickly as 3 months. The risk management process is standard to that of the DoD Risk Management processes (US DoD 2006). In figure 7, we have taken the DRRS-A risk data that is currently available and binned them according to our life cycle model (US DoD Product Management 2008).

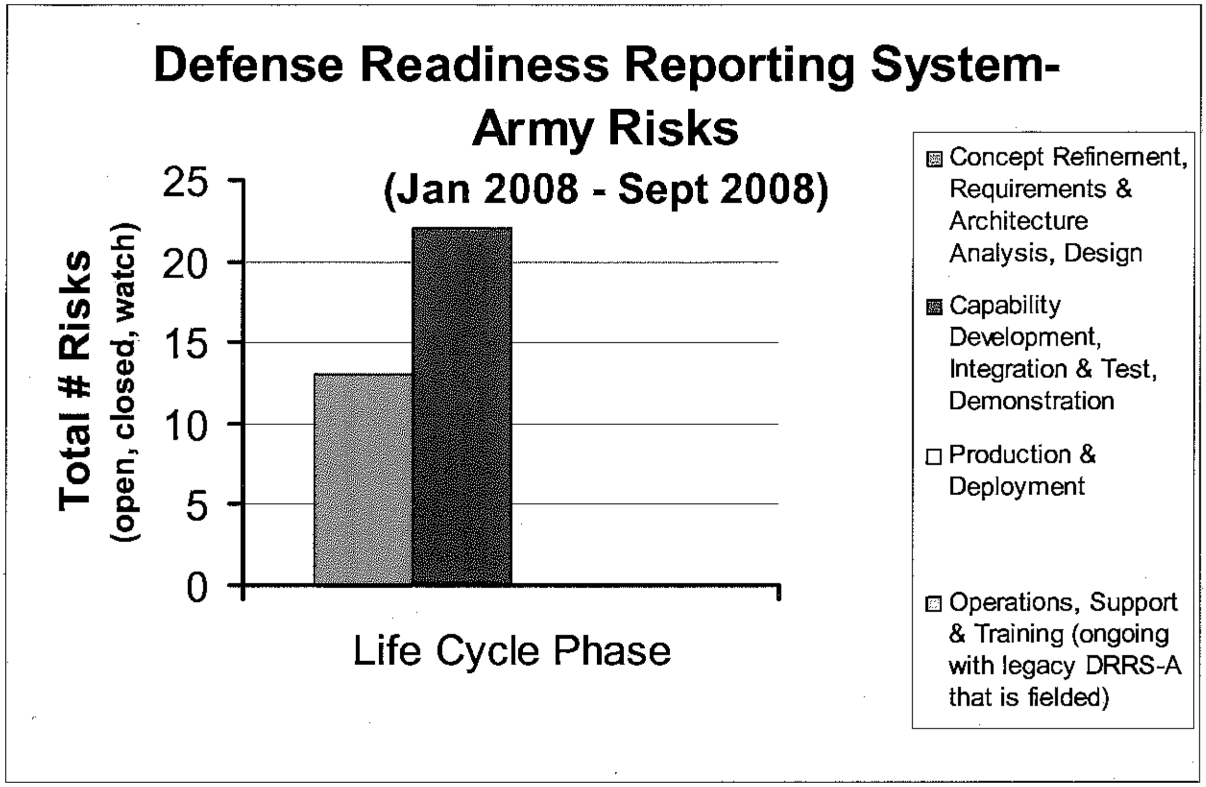

Figure 7. DRRS-A Risk Profile.

So far we have assessed that 63% of the program's risks are in the capability development, integration, test, and demonstration life cycle phase. Other risks were notably in the first phase, concept refinement, making up 36% risks captured. So far we have no documented risks in the production and deployment phase or in the operations, support and training phase.

If we were to impose the ARIEL model with our three identified phases, we could apply the risk management approach in an agile fashion throughout the life cycle of the DRRS-A program. Phase 1 of the DRRS-A program would include conducting logistics planning and have a feedback loop that would allow risk assessment early and often in the process. A cyclical loop between functions in phase 1 would also allow DRRS-A management to conduct early risk management for program constraints that would allow risk identification and mitigations of the entire life cycle.

Phase 2 of ARIEL would allow the DRRS-A program to conduct risk management for interface development, integration and security which has been identified as a high risk area. This loop would allow any risks from interface integration changes that impact security to be known by conducting the risk management process in ARIEL.

The continuous agile loop in ARIEL's phase 3 approach allow risk management to be conducted in the production, deployment and operations phase in a loop that would allow the program to track technical training and support with the help desk. The DRRS-A program has a large user community and conducts help desk support throughout the month. The cyclical risk management process together with the agile approach of ARIEL would allow coordination between help desk and training support which keeps up to date status and close collaboration between help desk support and training classes and materials. The loop between phase 1 and 3 closes the cycle between primary functions and allows for close coordination and collaboration between users and stakeholders.

These findings reiterate our position that there is a need for integration of risk management throughout an agile systems engineering life cycle process to capture risks in every phase of a program. The standard guidelines for conducting risk management in a program needs an evolutionary approach to conform to the changing development processes of complex environments and systems and the approaches that are used to meet this goal. In this case study, the risk management was conducted heavily in the first two phases although the program executes a full life cycle. This also supports our findings that risk management is heavily executed during the software development phase of a program.

5. Conclusion and Future Works

Risk is inherent in every project and can lead to failures or major disruption which can cause insurmountable loss of time, money and consumer confidence. We have learned that most projects conduct risk management for the overall program or specifically in the software development phase leaving other life cycle phases of a program in jeopardy of conducting risk assessments and mitigations. Changing environment and complexity of systems need risk management to be dynamic and adaptive especially during agile systems engineering life cycle processes and phases. Also, with new life cycle models emerging, risk management processes should be incorporated throughout the cycle and not just the beginning and in development phases. Our framework and risk model appeal to program management to build risk management inherently throughout the agile life cycle process which will in itself lead to lower risk programs and higher successful outcomes.

Our research and analysis of risk management integrated in agile systems engineering life cycle provides managers and team leads elegant approaches without confinement of heavy process and strict standards but yet allows for mandatory risk managility in complex environments and systems. Managing risks throughout the process instils redundant behaviour within and in between life cycle phases that allow risks to be assessed more often in which the likelihood of the risks happening decreases.

Future works will include categorizations of types of risk areas in agile systems engineering life cycle and where in the life cycle phase they generally appear. In conclusion of this research, we found that most risks are conducted during the life cycle phase so further research would go to understanding this behaviour and functions grouped into this phase (integration, test and demonstration) and conclude if further decomposition or a break out phase is necessary for integration of risk management. Our future works will also include integrating risk management in agile systems engineering life cycles to assess risk patterns and analyze where they are most apparent. Methodologies will be developed along with approaches to integrating risk management in agile systems engineering life cycles.

6. References

1. Boehm, B. 1991. Software Risk Management: Principles and Practices. IEEE Software, 8(1):32-41.
2. Boehm, B. et al. 1988. Using the Win-Win Spiral Model: A Case Study. IEEE Computer, 31(7), 33-44.
3. Boehm, B., and R. Turner. 2004. Balancing Agility and Discipline: A Guide for the Perplexed. Addison-Wesley: Boston.
4. Boehm, B., and W. Hansen. 2001. The Spiral Model as a Tool for Evolutionary Acquisition. *CrossTalk* 8-10.
5. Cherinka, R.; Mathews, R.; Miller, R.; Pitcher, D.; Sears, R.; Semanchik, T. 2007. Agile Capability Development, Assessment and Transition in Support of the Global War on Terror (GWOT). *IEEE*. June.
6. Conrow, Edmund. 2005. Risk Management for Systems of Systems. *CrossTalk* Feb: 1-4
7. Hansen, W.; Foreman, J.; Albert, C.;Axelband, E.; Brownsword, L.; Forrester, E. 2001.Spiral Development and Evolutionary Acquisition: The SEI-CSE Workshop. Software Engineering Institute, Carnegie Mellon University, Special Report, May.
8. IEEE Standard 1540. 2001. IEEE 1540 Standard for Lifecycle Processes-Risk Management. Institute of Electrical and Electronics Engineers, Inc. New York, NY.
9. IEEE Standard 12207.0. 1998. IEEE Standard for Information Technology Software life cycle processes, Institute of Electrical and Electronics Engineers, Inc. New York, NY.
10. INCOSE. 2004. Systems Engineering. http://www.incose.org/practice/whatissystemseng.aspx
11. Karduck, Achim.; Sienou, A.; Lamine, E,; Pingaud, H. 2007. Collaborative Process Driven Risk Management for Enterprise Agility. IEEE International Conference on Digital Ecosystems and Technologies.
12. Lorell, Mark,; Lowell. J,; Younossi O. 2006. Evolutionary Acquisition-Implementation Challenges for Defense Space Programs. RAND. Santa Monica, CA.
13. Merriam-Webster Dictionary Online. 2008. Risk. http://www.merriam-webster.com/dictionary/risk.
14. Military Army News Services. 2006. Army Unit Status Reports Move Online, http://www.globalsecurity.org/military/library/news/2006/03/mil-060306-arnews03.htm
15. Nyfjord, Jaana and M. Kajko-Mattsson. 2007. Commonalities in Risk Management and Agile Process Models, IEEE- Conferences on Software Engineering Advances.
16. Project Management Institute. 2004. A Guide to the Project Management Body of Knowledge (PMBoK), 3rd Ed. ANSI/PMI 99-001-2004, Project Management Institute, Newton Square, PA.
17. Raduege, Lt. Gen. Harry D. 2008. Managing Enterprise Risk in a Risky World, *Signal* (AFCEA International Journal) October Issue: 108.
18. Sommerville, I. 2006. Software Engineering, 7th Ed. Addison-Wesley, Reading, MA.
19. Turner, Richard. 2007. Toward Agile Systems Engineering Processes, *CrossTalk* April.

20. U.S. Department of Defense. 2003. Department of Defense Directive 5000.1: The Defense Acquisition System. Washington, DC: Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics.
21. ———. 2006. Risk Management Guide for DoD Acquisition, 6th ed. Vers.1.0, OUSD(AT&L) Systems and Software Engineering/Enterprise Development. http://www.acq.osd.mil/sse/ed/docs/2006-RM-Guide-4Aug06-final-version.pdf
22. ———. 2007. Defense Acquisition Guide: The Defense Acquisition Management Framework. Washington, DC: Office of the Under Secretary of Defense for Acquisition, Technology, and Logisitics. https://akss.dau.mil/dag/DoD5002/Figure1.asp
23. ———. 2008. Systems Engineering Guide for System of Systems, Version.1.0. May.
24. ———. 2008. Product Management Office Strategic Battle Command Defense Readiness Reporting System-Army Risk Management Data Collection, Version 2.0. September.
25. Verman, Kunal. 2007. Project Management Challenges and Best Practices for Enterprise Packaged Applications, *PMWorldToday*, Vol. IX, Issue VIII, Bangalore.
26. Williams R. et al. 1999. Software Risk Evaluation (SRE) Method Description,Version 2.0. Technical Report, CMU/SEI-99-TR-029, SEI/CMU, Pittsburg, PA.

7. Author Biographies

Dr. Ali Mostashari

Dr. Ali Mostashari is the Director of the Complex Adaptive Socio-technological Systems (COMPASS) Research Center and an Associate Professor of Systems Engineering at the School of Systems and Enterprises at Stevens Institute of Technology. His research focuses on the application of complexity science to systems engineering, and to socio-technological systems analysis and design. He currently serves as a senior strategy consultant for the United Nations Development Program's Knowledge Management 2.0 strategy. Dr. Mostashari was selected as a Asia 21 Young Leader, and was nominated by the UNDP Assistant Secretary General for Africa for the World Economic Forum's Young Global Leaders 2008 award. In 2004, he was selected as a top finalist of UNDP's Leadership Development Program from over 7000 applicants from 78 countries worldwide. Between 2004 and 2007 he served as a LEAD Project Manager and Strategic Resource Manager, with over $2 billion in development project portfolio in sub-Saharan Africa. Dr. Mostashari holds a Ph.D. in Engineering Systems from MIT, a Master's in Civil Engineering from MIT, a Master's in Technology and Policy from MIT, a Master's in Chemical Engineering from the University of Nebraska and a Bachelor's in Chemical Engineering from Sharif University of Technology.

Dr. Mohammad Mansouri

Dr. Mohammad Mansouri is a Postdoctoral Associate at the Center for Adaptive Socio-technological Systems (COMPASS) of the School of Systems and Enterprises (SSE) at Stevens Institute of Technology. He is conducting research on resilience of ports, maritime transportation and infrastructure systems, and enterprises. Dr. Mansouri received his Doctor of Science in Engineering Management from The George Washington University and for several years served as a management consultant for The World Bank, HAND Foundation, NIAC and other for-profit and non-profit organizations on issues of development planning, strategic management, and decision-making.

Dr. Robert Cloutier

Rob Cloutier is an Associate Professor of systems engineering in the School of Systems and Enterprises at Stevens Institute of Technology. He has over 20 years experience in systems engineering & architecting, software engineering, and project management in both commercial and defense industries. Industry roles included lead avionics engineer, chief enterprise architect, lead software engineer, and system architect on a number of efforts and proposals. His research interests include model based systems engineering and systems architecting using UML/SysML, reference architectures, systems engineering patterns, and architecture management. Rob holds a BS from the US Naval Academy, an MBA from Eastern College, and his Ph.D. in Systems Engineering from Stevens Institute of Technology.

Ms. Portia Crowe

Ms. Portia Crowe is currently a PhD student at the School of Systems Engineering at Stevens Institute of Technology in Hoboken, NJ. She holds a Bachelors of Science Degree in Computer Science from Rutgers University and a Masters of Science Degree in Engineering Management from New Jersey Institute of Technology. She is conducting research in risk management and agile systems engineering. She currently works for the Department of Defense on several agile projects to include US Army readiness systems and command and control (C2) systems to include Army Battle Command Systems (ABCS). Portia has received two Commander's Award for Civilian Service; one for successful implementation of the Army readiness reporting program and another for excellence in research and development of Army technology objectives.