AGILE AND LEAN PRINCIPLES AND SYSTEMS ENGINEERING – A SYNERGY?

S. M. Joseph-Malherbe[1]
[1]CSIR, South Africa
sjoseph@csir.co.za

ABSTRACT

Systems engineering is a generally accepted methodology in military development. Systems engineering is an interdisciplinary approach to realize successful systems. In most commercial organizations systems engineering is not an accepted or mandated methodology, however methodologies such as Agile and Lean are utilized to deliver successful systems. Due to the phenomenal rate of technology advancement as well as customers demanding quicker delivery of solutions, systems engineering is at times not efficient and effective enough to support rapid fielding of products. The objective of this piece of work is multi-fold: (1) to explore and present fundamental Systems Engineering, Agile and Lean principles (2) the evolution or lack of Lean and Agile in systems engineering and (3) recommendations for further research.

1. INTRODUCTION

Systems engineering has emerged as an effective way to manage complexity and change in large, complex development projects. It is an accepted and client-mandated methodology for large complex systems development in the military and aerospace industries. In most commercial enterprises systems engineering is not an accepted or mandated methodology, however other methodologies such as Agile and Lean are utilized to deliver successful systems.

Systems engineering adds immense value for a system with well-defined objectives, that is, systems requirements that are well understood. There are various challenges that face systems engineering. One such challenge is when developing a system without well-defined objectives. Exponential technology evolution makes it important in the military context to deliver a solution before a technology becomes sub-optimal or no longer relevant to the intended use. Systems engineering is at times not efficient and effective enough to support the need to get products fielded as quickly as possible. Therefore it is necessary to enhance systems engineering by exploring methodologies that have been successful in other industries.

The purpose of the paper is to introduce the fundamental concepts of Systems Engineering, Agile and Lean principles as well as to present the current research in applying Agile and Lean to systems engineering.

The remainder of the paper is organized as follows. In Section 2, we introduce definitions to ensure clarity and consistency of the use of terms. In Section 3, 4 and 5, the fundamentals of the three areas of interest, Systems Engineering, Agile and Lean is introduced as well as the current research in applying Agile and Lean to the existing body of knowledge in systems engineering. Section 6 is a discussion on some areas of synergy and the expected impact and changes required in systems engineering. Lastly, in Section 7, a summary is provided with an indication of the direction of future research.

2. DEFINITIONS

Words and phrases like *methodology*, *process*, *methods*, *tools* and *system life cycle* are widely used and understood. The specific interpretation of each word varies with the context it is used in. Definitions are given in Table 1 to clarify these terms for their use in this paper.

Process: A logical sequence of tasks performed to achieve a particular objective. A process defines "*what*" is to be done, without specifying "*how*" each task is performed. The structure of a process provides several levels of aggregation to allow analysis and definition to be done at various levels of detail to support different decision-making needs [8].
Method: Consists of techniques for performing a task. It defines the "how" of each task (words "method", "technique", "practice" and "procedure" are often used interchangeably). At any level, process tasks are performed using methods. What is important to note is that each method is also a process itself, with a sequence of tasks to be performed for that particular method, therefore, the "how" at one level of abstraction becomes the "what" at the next lower level [8].
Tool: It is an instrument that, when applied to a particular method, can enhance the efficiency of the task. The purpose of the tool is to facilitate the accomplishment of the "*how*s" and in a broader sense, a tool enhances the "*what*" and the "*how*" [8].
Methodology: A collection of related processes, methods and tools. It can be thought of as a recipe – an application of related processes, methods and tools to a class of problems [8].
System life cycle: Commonly, used to refer to the stepwise evolution of a new system from concept through development and on to production, operation and disposal [9].

Table 1: Definitions

Life cycle models break up a system life into stages where transition from one stage to the next serves as major milestones and agreed methods specific to a discipline contribute to the realization of the system in each stage.

3. SYSTEMS ENGINEERING

In the development of large complex systems, there are several different processes involved, the system life cycle process, the systems engineering process, the acquisition process, the design engineering process, project processes and the manufacturing process to name a few. In this context, the systems engineering process refers to the process applied to the total system development.

Systems engineering consists of two sub-processes, a technical management process and the technical process. The technical management process can be further broken down into, project planning, review and re-planning and change control. This sub-process is an integral part of project management. The technical process enables the coordination of interaction between engineering specialists, stakeholders, end-users and manufacturing. Systems engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process considering both the business and the technical needs of all stakeholders [1].

Systems engineering [SE] as defined by INCOSE is an interdisciplinary approach and means to enable the realization of successful systems [1]. In other words, the function of SE is to guide the engineering of complex systems [9]. It focuses on defining customer needs and required functionality early in the development cycle, documenting the requirements then proceeding with design synthesis and system validation while considering the complete problem [1].

SE addresses the system as a whole allowing views of the system from the outside, the system interfaces, interactions with other systems and the environment, as well as the inside of the system [9]. The promise of SE is "to provide better systems in less time and cost [sic] with less risk" [5].

The evolution of a system from concept through development, on to production, operation and disposal is the system life cycle. The system life cycle has three aspects: the business aspect, the budget aspect, and the technical aspect [1] – the overall effort has to ensure that these three aspects are in balance through the life cycle. The system life cycle is defined to establish a framework to meet the stakeholders' needs. This is done by defining life cycle stages and using decision gates to determine the readiness (transition criteria) to move from one stage to the next. Transition criteria include completion criteria for the current stage as well as choice and entrance criteria for the next stage. The role of systems engineering changes as the engineering effort moves through the system life cycle. A generic life cycle, see Figure 1, consists of the following stages:

- Exploratory research – identify stakeholders' needs and explore ideas and technologies.
- Concept – define the stakeholders' needs, explore feasible concepts and propose viable solutions.
- Development – refine system requirements, create the solution description, build, verify and validate the system.
- Production – the system is produced, inspected and verified.
- Utilization & support – operate the system to satisfy customers' needs and provide sustained system capability.
- Retirement – the system is stored, archived or disposed.

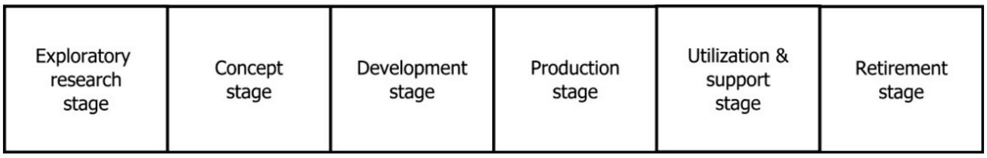

Figure 1: Generic system life cycle

The purpose of each stage and the approaches to realize the output in each stage varies. Although the problems addressed in each stage of the system life cycle are unique to that stage, the SE principles and their application (the systems engineering method) in each stage are the similar. The effort required in each step of the SE method varies depending on the type of system and the development stage. The SE method consists of the following basic activities:

- Requirements analysis - problem definition.
- Functional definition - functional analysis and allocation.
- Physical definition - synthesis, physical analysis and allocation.
- Design validation - verification and evaluation).

The SE method transforms needs and requirements into a set of system descriptions (adding more detail in each level of development), makes information available for decision makers (such as specifications) and provides inputs to the next level of development. The well-known "football diagram" by Project Performance International is an example of the SE method [17].

4. WHAT IS AGILE?

In February 2001, seventeen software developers experienced in various programming methods met to outline the values and principles that will allow teams to develop quickly and to respond to change. The result is the Agile *manifesto* and *principles* which form the foundation of the agile movement. The manifesto encourages better ways of developing software and based on the manifesto a collection of principles define the criteria for agile software development processes.

4.1. Agile Manifesto and Principles

The manifesto defines four values [19]:

- **Individuals and interactions** over processes and tools.
- **Working software** over comprehensive documentation.
- **Customer collaboration** over contract negotiation.
- **Responding to change** over following a plan.

The manifesto states that while value should be given to the concepts on the right of the value statement, the concepts on the left hand side should be valued even more. For example, in the first value people are the most important ingredient of success. A good process will not save a project from failure if the team is not strong but a bad process can make even the strongest team ineffective. Building the project team is more important than building the project environment, that is, rather work to create the team and then let the team change the processes and tools as required.

According to the Agile Alliance, the following principles underlie the Agile manifesto [19]:

- The highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development.
- Deliver working software frequently.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals and give individuals the environment and the support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity is essential.
- The best architectures, requirements and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

Numerous software development methods fall under the umbrella of Agile these include SCRUM, Crystal, Feature Driven Development, Adaptive Software Development, Dynamic Systems Development Method and Extreme Programming. Extreme Programming focuses on the software developer and methods like SCRUM focus on managerial processes [11].

These methods address changing requirements in the software engineering environment. Fundamentally, there are two scenarios for which software is developed; on the one hand software is developed for an organization like NASA, where the software development cycle is predictable – it requires a lot of time, documentation, large teams and the requirements are stable. On the other hand software is developed for business where the value of software features change rapidly – what may have been a good set of requirements becomes out-dated in a few months and user interfaces are a typical example. In the second scenario change is inherent in software development, and as a result so is uncertainty. The intangible nature of software makes it difficult to see what value a software feature will add until the customer uses it. It therefore becomes necessary to make available early versions of the software to really understand which features are valuable to the customer and which ones not.

When applying Agile, testing is integrated throughout the life cycle, features are completed one by one and the user is actively involved in the development in order to see the product regularly while being able to direct the development. This approach ensures visibility of progress and provides an accurate measure of the product's completeness on a regular basis. The heart of these methods is iterative and incremental development. Each cycle is a short period, during which a subset of the system's requirements is analyzed, designed, coded and tested resulting in the delivery of production-quality code. Incremental development divides the capabilities of a system into subsets and then develops and delivers each subset of requirements per iteration [10]. Figure 2 is an example of an Agile method, the *Disciplined Agility Process – the basic model*.

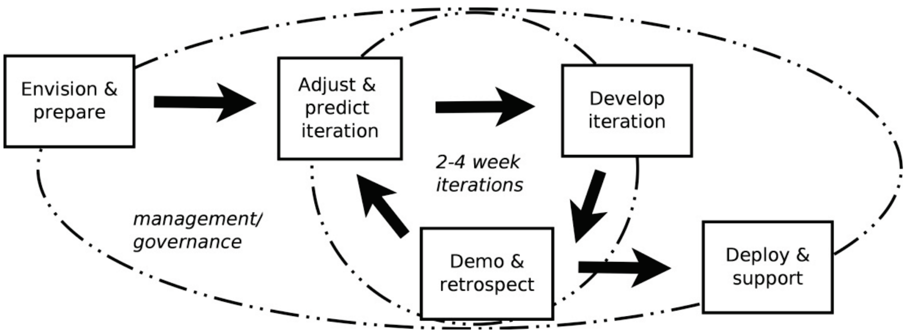

Figure 2: Disciplined Agility process – the basic model [4]

4.2. Agile System Engineering

A distinction is made between Agile *systems-engineering* and *Agile-systems* engineering by Habelfellner and de Weck [8]. Agility is defined as a property of a system that can be changed rapidly. Agility may be realized both in the process of engineering systems (Agile *systems-engineering)* and in the resulting system (*Agile-systems* engineering).

Turner [4] talks about introducing agility into the process of engineering systems and suggests some characteristics of agility can be introduced in the systems engineering process:

- *Learning attitude*: Take advantage of lessons learned and adapt both processes and systems to meet customer needs.
- *Focus on value to customer*: Customer prioritizes requirements and progress is measured by operational features.
- *Short iterations delivering value*: The goal of each release is a working system. It is a risk-driven, reality based iteration planning.
- *Neutrality to change*: Change is seen as inevitable. Design processes and systems for change.
- *Continuous integration and test-driven development*: Tests are written first, that is, requirements are defined by the tests that validate them.
- *Lean attitude*: The removal of low value or unnecessary activities.

- *Team ownership*: The development team has primary responsibility and authority over its own plans and processes and quality and performance is everyone's responsibility.

Agile methods appear to add the most value to the customer in software development and these methods need to be considered in systems engineering in which systems are becoming more software-intensive and traditional approaches do not fully aid in realizing successful systems. INCOSE adapted the twelve Agile principles for systems engineering by referring to *software* in each of the principles to *software and other system elements* [1].

5. WHAT IS LEAN?

In 1988, a group of researchers from Massachussets Institute of Technology (MIT) led by Dr. James P. Womack were examining the international automotive industry. They observed what was happening at the Toyota Motor Company, the Toyota Production System (TPS), and struggled with a term to describe what they were seeing compared to the traditional mass production. They saw that Toyota [12]:

- Needed less effort to design, make and service their products.
- Required less investment to achieve a given level of production capacity.
- Produced products with fewer defects.
- Used fewer suppliers.
- Performed its key processes in less time and with less effort.
- Needed less inventory at every step.
- Had fewer employee injuries.

They described what they saw as Lean production in contrast to mass production. In 1990 James Womack, Daniel Jones and Daniel Roos documented the experience of Lean production in the book, *The Machine that Changed the World* which grew from a five year project investigating the role of the automobile industry in the world economy. In 1996 James Womack and Daniel Jones wrote the book *Lean Thinking* which generalizes the lessons from the original project and proposed that Lean encompasses all aspects of product creation and delivery, but they concentrated on factory floor examples.

The Lean Enterprise Institute [13] states that the core idea behind Lean is to maximize customer value while minimizing waste, creating more value for customers with fewer resources. Lean may be viewed as a continuous improvement methodology. In *Lean Thinking*, the process of creating value without waste is captured into five principles of Lean:

- Define **value** from the perspective of the end customer. Value is expressed in terms of how the product meets the customer's needs at a specific time and price.
- Identify the **value stream** for each service, product or product family and eliminate waste. The value stream is a sequence of activities that contribute to the value identified, thereafter the activities that do not contribute value to the product or service is deemed to be necessary or not.
- Make the value stream activities **flow**, the uninterrupted movement of product or service through the system to the customer.
- Have the customer **pull** from the stream. Once flow is established, allow the customer to pull product or service through the process when the customer needs it.
- Pursue **perfection**. The process begins again in constant attempt to remove waste, improve flow and meet customer needs.

Over time a sixth Lean principle was added, r**espect for people**, which encompasses providing an environment of mutual respect, trust and cooperation. These Lean principles are depicted in Figure 3.

Categories of waste are listed and at times an eighth waste category is added to the original seven. For example, an addition by Liker [15] is *unused employee creativity* referring to losing time, ideas, skills, improvements and learning opportunities by not engaging or listening to employees. The seven original categories of waste are:

- Transport – moving material or information from place to place.
- Unnecessary inventory – all components, work in process and finished product not being processed.
- Unnecessary motion – movement by people or equipment that is more than required.
- Waiting – Waiting for materials, information or decisions.
- Over production – producing more or ahead of demand.
- Over processing – performing unnecessary processing on a task or an unnecessary task.
- Defects – the effort involved in inspecting for and fixing defects.

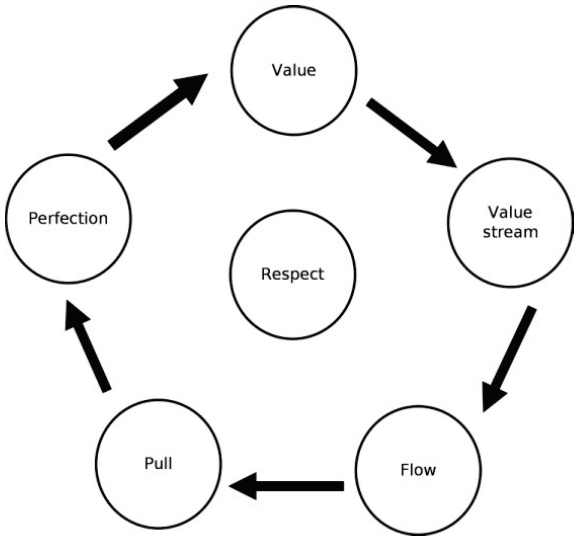

Figure 3: Lean Principles

Nave [14] states that assumptions are made when employing continuous improvement methodologies to improve processes. These assumptions are that the design of the product or service is correct, the design of the product or service is the most economical, the product or service fulfills the requirements of the customer and the management structure supports and nourishes change. At times, these assumptions are not valid and further analysis is required prior to improving just the process.

Although Lean has been mostly associated with manufacturing and production processes, Lean methods cover the total enterprise which includes all aspects of operations. Lean is the search for perfection through the elimination of waste and the insertion of practices that contribute to reduction in cost and schedule while improving performance of products [6]. Over time, the term “lean” became associated with enterprises with a certain capability – the ability to accomplish more with less.

Going back to the beginning of this section and the Toyota Production System, through further observation and interactions with Toyota, it came to light that Toyota achieves high performance in product development through both efficient practices (TPS) and a culture of learning and continuous improvement [16]. Liker [15] states that the success of implementing TPS is rooted in Toyota's philosophy which is based on an understanding of people and what motivates them.

The Lean Aerospace Initiative (LAI), who has been producing research in support of the military and industry on the topic of lean and its benefits, collaborated with INCOSE and produced the emerging body of knowledge called Lean Systems Engineering (LSE).

5.1. Lean Systems Engineering

So what is Lean Systems Engineering? SE grew out of the space industry in response to the need to deliver technically complex systems that worked flawlessly. SE emphasized technical performance and risk management of complex systems. Lean thinking grew out of the automotive industry in response to the need to deliver quality products with the minimum use of resources. Lean therefore emphasized waste minimization and flexibility in the production of high quality affordable products with short development and production lead times. Both methodologies evolved over time with the common goal of delivering system life cycle value to the customer. Lean SE is the area of synergy of Lean and SE. It applies the fundamentals of Lean thinking to SE with the objective of delivering the best life cycle value for complex systems with minimum waste.

When applying Lean principles to SE, the process is similar to the process depicted in Figure 3. The process starts by considering how the customer defines ***value*** in products and processes, then describing the ***value stream*** for creating products and services, optimizing the ***flow*** through the value stream and eliminate waste, encouraging ***pull*** from the value stream and striving to ***perfect*** the value stream to maximize value to the customer. All these activities should be conducted on a foundation of ***respect*** for customers, stakeholders and project team members [1].

Lean Enablers for SE (LEfSE) is a collection of practices and recommendations of SE based on Lean thinking and it is organized into the six Lean principles. The practices cover SE and other relevant management practices. The full text of LEfSE is available online on the INCOSE website and an explanation of the enablers is available in [8]. Each Lean principle in the context of LEfSE is explained as well as the purpose of the enablers for each principle [1]:

- **Value** - The enablers promote a process of establishing the value of the end-product or system to the customer, the process should be customer focused, involving the customer frequently as well as aligning the project team accordingly.
- **Value stream** - The enablers emphasize waste-preventing measures. Systems engineers should prepare for and plan all end-to-end linked actions and processes necessary to realize streamlined value, after eliminating waste.
- The **flow** principle lists enablers that promote the smooth flow of quality work. The enablers to aid the flow include frequent clarification of the requirements, frequent opportunities for decision making, making progress visible to all, using the most appropriate tools, communication and coordination methods.
- **Pull** promotes proactive coordination of between employees to eliminate the waste of misunderstanding, defects, rework and waiting.
- The **perfection** principle strives for excellence and continuous improvement of the SE process and related enterprise management. The enablers support best means of communication, coordination and collaboration to enable continuous improvement. This principle elevates the role of the lead systems engineer to lead and integrate the technical team from start to finish

- **Respect** for people contains enablers that promote the enterprise culture of trust, openness, honesty, respect, empowerment, cooperation, teamwork, synergy, good communication and coordination and enable people for excellence.

Lean SE recognizes people as the most valuable contributors to an engineering effort and as a result there is emphasis on communication, coordination and learning activities. LEfSE is not intended to be a mandatory tool rather it should be used as a reference for good practices.

6. DISCUSSION

6.1. Agile and Lean in SE

Traditionally systems engineering is approached assuming that all the stakeholders have perfect knowledge of the system at the beginning of the project. That is, we are able to define requirements that are feasible, verifiable and current to completely describe the system; decompose these requirements to complete specifications and at the same time accurately estimate the resources required to realize the effort; schedule the work and monitor and measure the progress using a life cycle model. However, it is not realistic to assume absolute knowledge of the system at the beginning of a development. One forgets that we are here to “engineer” the system, to design and build, to figure out the “how” and this by nature adds uncertainty when engineering systems.

In addition, systems engineering assumes that the customer is fully understood through the predefined and agreed upon requirements and operational concepts early in development. When considering interactive end-user applications the value of a feature can only be determined once it is evaluated by the end-user and not at any time prior. In addition to this the developer only has partial knowledge about the end-user expectations that is expressed explicitly and has limited or no access to the tacit knowledge. The way tacit knowledge can be shared is through socializing by including more interactions with the end-user throughout the system development process.

If the two assumptions discussed above are not valid there will be a negative impact on the successful realization of a system. The application of Agile and Lean approaches to SE development will assist in delivering the best value to meet the customer needs especially in light of customers demanding quicker delivery due to dynamic and uncertain market conditions and the rapid evolution of technology-enabled systems.

There are numerous advantages in employing Lean and Agile methods in SE. The utility in adopting these methods lies in the lower cost to realize a solution, the project being delivered within schedule and improving the quality of the product or service. The quality of the product or service can be increased by creating more opportunities for interaction with the customer through which the concept and development of the system is refined to meet their need and expectations. An advantage for an organization lies in its ability to respond and adjust more rapidly to the changing environmental conditions. The aerospace industry has experienced benefits not only from applying lean principles in manufacturing and supply chain management but also from applying in to the engineering effort [18]. In addition, Agile methods have had influence in the military development environment due to contractors on projects adopting the methods whilst developing a system for the military [11].

INCOSE has adapted these methods within the SE framework which is available in the INCOSE handbook. For example, the role of Agile in SE is captured in an enabler of the ***value*** principle in LSE - “*develop an agile process to anticipate, accommodate and* *communicate changing customer requirements* **[8]**". Adopting a method for large, complex development projects requires changes in the traditional approaches by all involved. Lean and Agile methods for example require close and regular involvement of the *real* end-users in the development process. Typically, the acquisition organization speaks for the end-user and they attempt to have end-user inputs but the value and quality of the input is dependent on end-user availability and numerous other issues. This requirement for regular involvement requires changes in the traditional ways the end-user, the contracting agency and the development agency interacted with one another and changes include employing collaborative capabilities like wikis.

Also, the traditional approach to wait till the system is ready to be released to test is challenged by the need for testing throughout the project when adopting Agile methods. One also needs to determine how the standard milestone criteria in the acquisition life cycle employed in SE such as critical design reviews will be met.

In general, enterprise wide processes will be influenced when utilizing either or both methods and there needs to be a conscientious effort by all stakeholders involved in the development to plan for and anticipate the changes needed. Most of all, training and support is required to ensure adequate understanding of Lean and Agile.

6.2. What does all this mean?

Concerted effort is required in developing/tailoring processes to determine appropriateness in employing a method to fit a situation and context. An approach to successfully incorporate Agile and Lean methods is to define a technical development strategy in each stage of the system life cycle. A technical development strategy may consist of development methods, a development strategy, a delivery strategy as well as a testing strategy [2]. A development strategy can be either incremental or evolutionary and the system can be produced in single or multiple deliveries. Development methods such as Waterfall, Spiral, Vee and Agile are utilized in defining the start, stop and process activities appropriate to the life cycle stages of the system life cycle [1]. The primary focus of a development method is on how to navigate through each stage. A testing strategy addresses testing early in development, testing later in the development or testing as one develops.

Developing a technical development strategy in each stage enables one to better bridge the gap between the systems engineering activities and other activities. In each stage one may employ various development methods to address the various engineering disciplines, for example, an agile method may be employed for the design and development of the software component and the waterfall method for the design and development of the mechanical component and the SE method for the system level descriptions.

The proposal to develop a technical development strategy framework for each stage of the system life cycle is a means to manage and coordinate the technical effort when employing Lean, Agile and SE methods. Further research is required to fully address the feasibility of such a proposal.

7. CONCLUSIONS AND FURTHER RESEARCH

As systems grow more complex, new ways of dealing with abstraction, concurrency and uncertainty need to be developed. There is no silver bullet, however it is not unreasonable to be aware and accept that there are regular bullets available [4]. Adopting new methods and philosophies and integrating them into current SE practices can significantly improve the capability of engineering systems.

The intent with this paper is to explore Agile and Lean principles and to present areas of synergy with systems engineering as shown in literature. At this juncture, the next step in the research agenda is to explore and understand the feasibility in developing a technical development strategy framework for each stage of the system life cycle to incorporate Agile and Lean principles and methods in addition to the traditional systems engineering approaches. The technical development strategy will consist of development methods, a development strategy, a delivery strategy and a testing strategy. Further research will also include understanding the impact of adopting these methods on the acquisition life cycle process.

8. REFERENCES

[1] **INCOSE.** 2010. *Systems Engineering Handbook v.3.2*, International Council on Systems Engineering. http://www.incose.org.

[2] **Forsberg, K and Mooz, H.** 2001. A Visual Explanation of Development Methods and Strategies including the Waterfall, Spiral, Vee, Vee+, and Vee++ Models, *The Centre for Systems Management*.

[3] **Boehm, B.** 1988. A Spiral Model of Software Development and Enhancement, *IEEE Computer*, 21(5), pp 61 – 72.

[4] **Turner, R.** 2007. Toward Agile Systems Engineering Process. *CrossTalk, The Journal of Defense Software Engineering*, pp 11 – 15.

[5] **Vanek, F., Jackson. P. and Grzybowski, R.** 2008. Systems Engineering Metrics and Applications in Product Development: A Critical Literature Review and Agenda for Further Research, *Wiley Periodicals Inc, Systems Engineering*, 11 (2), pp 107-124.

[6] **Walton, M.** 1999. Strategies for Lean Product Development, *The Lean Aerospace Initiative*, WP99-01-91.

[7] **Department of Defense**. 2001. *System Engineering Fundamentals*. Defense Acquisition University Press.

[8] **Oppenheim, B. W., Murman, E. M and Secor, D. A.** Lean Enablers for Systems Engineering, *Wiley Periodicals Inc, Systems Engineering*, 14 (1), pp 29-55.

[9] **Kossiakoff, A., Sweet, W. N., Seymour, S. J. and Biemer, S. M.** 2011. *Systems Engineering Principles and Practice*, 2nd Edition, John Wiley & Sons

[10] **Leffingwell, D. and Muirhead, D**. 2004. Tactical Management of Agile Development: Achieving Competitive Advantage, *Rally Software Development Corporation White Paper*.

[11] **Lapham, M. A.** 2010. Considerations for Using Agile in DoD Acquisition, *Acquisition Support Program – Air Force*. http://www.sei.cmu.edu.

[12] **Sayer, N. J. and Williams, B.** 2007. *Lean for Dummies*. Wiley.

[13] **Lean Enterpise Institute,** http:/www.lean.org.

[14] **Nave, D.** 2002. How to Compare Six Sigma, Lean and the Theory of Constraints. *Quality Process Magazine*.

[15] **Liker, J.** 2004. *The Toyota Way*, McGraw-Hill.

[16] **McManus, H.L., Haggerty, A. and E. Murman.** 2007. Lean Engineering: A Framework for Doing the Right Thing Right, *The Aeronautical Journal*.

[17] **Halligan, R.** 2011. Model Based Systems Engineering – A New Methodology or an Old One in a New Jacket?, *Systems Engineering Newsletter*, SyEN #031.

[18] **Haggerty, A. and Murman, E.** 2006. Evidence of Lean Engineering in Aircraft Programs, *25th International Congress of the Aeronautical Sciences*.

[19] **Agile Alliance,** http://www.agilealliance.org.

[20] **Habelfellner, R. and de Weck, O**. 2005. Agile SYSTEMS ENGINEERING versus AGILE SYSTEMS engineering, *15th Annual International Symposium of INCOSE*.