Agile Systems Engineering

Lead Author: *Phyllis Marbach*

For a system, a life cycle usually starts at the concept definition phase, moves through stages until completion of this system, as defined in the concept definition stage. A model of the life cycle may be a physical, data or graphic representation of that life cycle. The process describes the steps to accomplish each stage of the life cycle including input to and output from this stage. Today's complex and increasingly highly connected systems face rapid obsolescence under the stress of technological change, environmental change, and rapidly evolving mission needs. For these systems to remain robust against disruption they must be architected to agilely adapt. To meet these needs, the system must be assessed to apply the process that best serves the system, subsystem or component of the system of interest (SOI).

It is important to determine the best life cycle to use for the SOI early in the concept definition phase. On a program that is going to operate agilely, especially if it will be a hybrid model with agile, and other life cycle models it is important to define and harmonize them at key integration points based on hardware or other long-lead item maturity. An informative case study where the key integration points were identified across multiple life cycles in use by various significant components of the South African Radio Astronomy Observatory (SARAO) provides an excellent example of a methodology tailoring process (Kusel 2020).

In the Agile SE process, the systems engineer works in an iterative, incremental manner, continually modeling, analyzing, developing, and trading options to bring the definition of the system solution into focus. An example

of this work will be analyzing and maintaining not only the requirements but also the architectural model of the higher-level requirements and the linkage from those high-level requirements to the analyzed lower-level requirements. In addition to the requirements and architecture representations, maintaining and verifying the interfaces are defined and followed as the development progresses are some of the systems engineers' tasks. These responsibilities of systems engineers are the same regardless of the life cycle, although sequencing and organization may be different. Program leadership, systems engineering, and all team members work in a culture that represents an Agile Mindset. The Agile Mindset is a combination of beliefs and actions from agile values that include focusing on delivering working capabilities often, trusting the knowledge workers to find the best solution, improving the product and process through regular demonstrations and retrospections, planning often to implement lessons learned.

□

Contents

Principles for Agile Development

Principles for agile development (Marbach 2015) provide a foundation for the working relationship across cross-functional teams that include Systems and Software Engineers working on customer projects that include both hardware development and software development. These principles, in Table 1, are a modified version that originated from the Agile Manifesto (Beck 2001) and were expanded to apply to systems engineering. Adopting these principles will enable teams of teams to produce high-value capabilities incrementally.

Principles for Agile Development (Marbach 2015)	SEBoK Traditional SE Principles

First, satisfy the customer through early and continuous delivery of valuable capabilities.	1. SE in application is specific to stakeholder needs, solution space, resulting system solution(s), and context throughout the system life cycle. 8. SE addresses stakeholder needs, taking into consideration budget, schedule, and technical needs, along with other expectations and constraints.
2. Plan for evolving requirements, and retain as much flexibility as is valuable throughout development; agile processes harness change for the customer, especially when change leads to a competitive advantage.	7. Stakeholder needs can change and must be accounted for over the system life cycle. 9. SE decisions are made under uncertainty accounting for risk.
3. Deliver working capabilities frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	5. The real physical system is the perfect model of the system. 2. SE has a holistic system view that includes the system elements and the interactions amongst themselves, the enabling systems, and the system environment.
4. Business personnel, customers or their advocates and implementers must work together daily throughout the project.	3. SE influences and is influenced by internal and external resources, and political, economic, social, technological, environmental, and legal factors. 13. SE integrates engineering disciplines in an effective manner. 14. SE is responsible for managing the discipline interactions within the organization.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	6. A focus of SE is a progressively deeper understanding of the interactions, sensitivities, and behaviors of the system, stakeholder needs, and its operational environment.

6. The most efficient and effective method of conveying information is personal conversation.	13. SE integrates engineering disciplines in an effective manner.
7. Working capabilities are the primary measure of progress.	5. The real physical system is the perfect model of the system.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.	8. SE addresses stakeholder needs, taking into consideration budget, schedule, and technical needs, along with other expectations and constraints. 9. SE decisions are made under uncertainty accounting for risk.
9. Continuous attention to technical excellence and good design enhances agility.	10. Decision quality depends on knowledge of the system, enabling system(s), and interoperating system(s) present in the decision-making process.
10. Simplicity “the art of maximizing the amount of work not done” is essential, especially within the implementation team. A truly agile development project does not force artificial reporting and process requirements on the implementation team.	4. Both policy and law must be properly understood to not over-constrain or under-constrain the system implementation.
11. The best architectures, requirements, and designs emerge from self-organizing teams, based on a minimal set of guiding principles.	10. Decision quality depends on knowledge of the system, enabling system(s), and interoperating system(s) present in the decision-making process.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.	4. Both policy and law must be properly understood to not over-constrain or under-constrain the system implementation.

The Principles for Agile Development emphasize focusing on working capabilities and keeping work in progress small. Table 1 maps the Principles for Agile Development to traditional SE Principles elaborated elsewhere in the SEBoK. The Principles for Agile Development are complementary to the traditional SE Principles and in some cases very similar. Click here for the complete set of the SEBoK Systems Engineering Principles.

Agile Systems Engineering Life Cycle Model for Mixed Discipline Engineering (Dove 2019) describes principles that facilitate operational agility in action: Sense, Respond and Evolve (SRE). A large program over the entire life cycle will benefit from applying these principles. In addition, the Scaled Agile Framework® (SAFe®) describes principles that facilitate developmental agility on large programs with teams of teams including systems engineering. The Operational Agility Principles and the SAFe Lean-Agile Principles are mapped in Table 2 to show the consistency between them. There is value in starting from an agile mindset and applying a set of principles.

Operational Agility Principles (Dove, et al)	**Scaled Agile Framework (SAFe) Lean-Agile Principles**
Sensing: External awareness (proactive alertness)	- Take an Economic view - Apply Systems Thinking
Sensing: Internal awareness (proactive alertness)	- Visualize and limit WIP, reduce batch sizes, and manage queue length
Sensing: Sense making (risk analysis, trade space analysis)	- Apply Systems Thinking
Responding: Decision making (timely, informed)	- Decentralize decision-making - Apply cadence, synchronize with cross-domain planning - Organize around value - Assume variability; preserve option
Responding: Action making (invoke/configure process activity to address the situation)	- Unlock the intrinsic motivation of knowledge workers - Base milestones on objective evaluation of working system
Responding: Action evaluation (verification and validation)	- Apply cadence, synchronize with cross-domain planning - Visualize and limit WIP, reduce batch sizes, and manage queue length - Base milestones on objective evaluation of working system
Evolving: Experimentation (variations on process ConOps)	- Base milestones on objective evaluation of working system

Evolving: Evaluation (internal and external judgement)	- Base milestones on objective evaluation of working system - Build incrementally with fast, integrated learning cycles
Evolving: Memory (evolving culture, response capabilities, and process ConOps)	- Assume variability; preserve option - Base milestones on objective evaluation of working system

Table 2. Comparison (or mapping) of Principles for cross-discipline teams. The SAFe Lean-Agile Principles are copyrighted material of Scaled Agile, Inc.

Using the Agile Mindset aligned with the Agile Values, the Principles of Agile Development and SAFe Lean-Agile Principles can be applied to the stages of the Agile SE Life Cycle.

A Generic Life Cycle Model shows a Definition stage, a Realization Stage and a Retirement Stage. Within each stage are further stages. Each SOI has a corresponding life cycle model which is composed of stages that are populated with processes. The processes within each stage may be the Vee, Iterative or Agile depending on the assessment of the SOI performed during the Definition stage. Several process models have been presented in literature and may be called Agile Systems Engineering Process, Model or Framework.

There are six system life cycle stages “commonly encountered” according to ISO/IES/IEEE 24748-1 Systems and Software Engineering – Life Cycle Management – Part 1: Guide for Life Cycle Management (ISO/IES/IEEE 2018). Dove describes an “asynchronous and concurrent life-cycle stage and process activity” that adds a seventh life-cycle stage, Situational Awareness (Dove 2019). This representation, Figure 1, was developed as part of four case studies of organizations who were using agile practices effectively in their systems development. Using this Agile Systems Engineering Life Cycle Model (ASELCM) the path may start at the “Concept” stage, move into the Situational Awareness phase, then move into the Development phase, back to the Situational Awareness phase and so forth around the circle. This situational awareness phase allows for a demonstration, review, and continuous improvement before shifting attention back to the next appropriate phase, such as back to concept or into development.

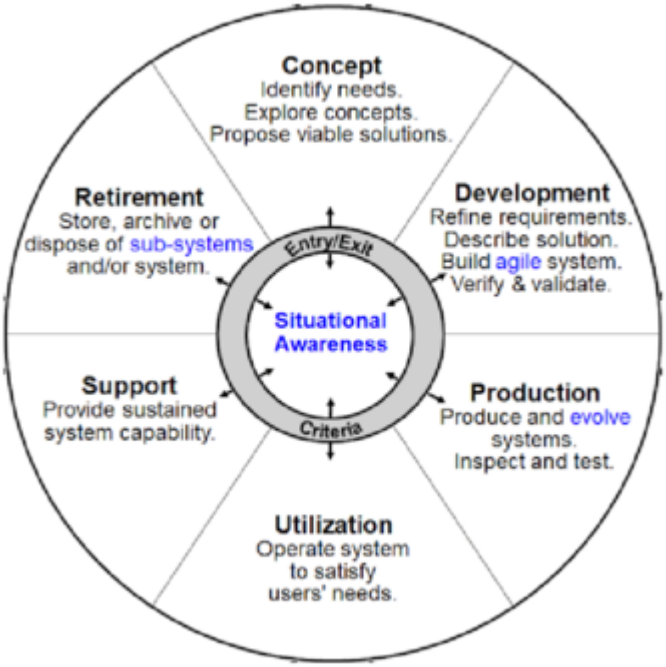

Figure 1. The Agile Systems Engineering Life Cycle Model (ASELCM). (Dove and Schindel 2019, Used with Permission)

The life cycle model represented in Figure 1 can be depicted visually using the S*Pattern Class Hierarchy shown in Figure 2. In this hierarchy the top level shows an entity relationship paradigm, "the minimum conceptual content necessary to model any system for purposes of engineering or science." (Schindel 2016). As one moves to the more specific model representations the ASELCM Pattern inherits content from parent patterns.

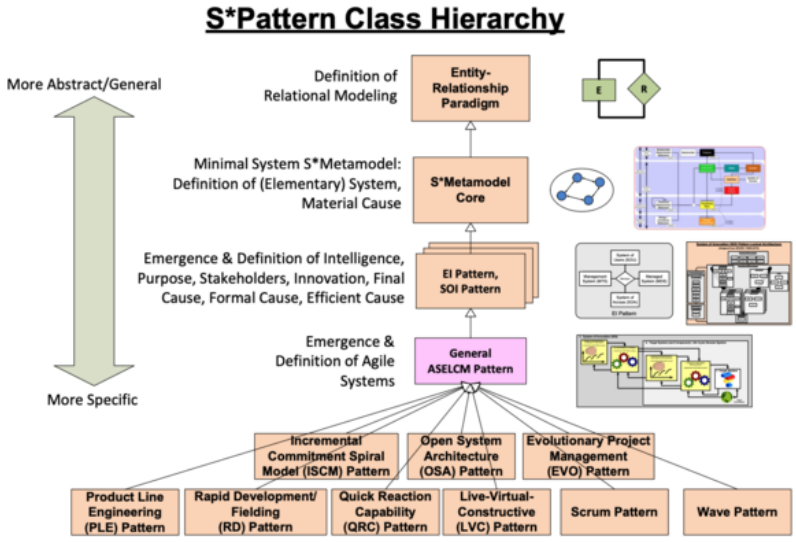

Figure 2. S* Pattern Class Hierarchy (Schindel and Dove 2016, Used with Permission)

The "General ASELCM Pattern" shown at about the middle of Figure 2 is enlarged in Figure 3. In developing

a systems of interest (SOI) one must consider not only the SOI we can think of as the target system, System 1 in Figure 3, but also consider the process that produces the target system, System 2 in Figure 3. Besides systems 1 and 2 there is a third system shown as the System of Innovation. This System 3 is the meta-view of that which influences the design, development, and operation of the target system, the embedded behavioral framework of System 1 and 2. In modeling the target system the other influencing systems 2 and 3 should be considered.

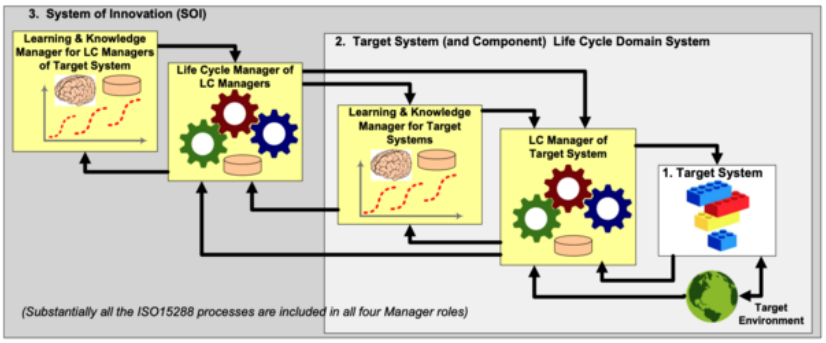

Figure 3. Agile Systems Engineering Life Cycle Model (ASELCM) Pattern (Schindel and Dove 2016, Used with Permission)

Other system engineering models, the Traditional (or waterfall), the Vee, Incremental, and spiral are described in those SEBoK articles.

Frameworks

The Agile Systems Engineering process steps that are performed in each of the stages often include:

1. Define the highest priority and/or highest risk item to work first, keeping design options open, until the last responsible moment. This produces a list of work items with the highest work item on top. This prioritized list of work items is called a program backlog.
2. During development iterations engineers analyze the requirement, design the solutions to meet those requirements, develop their products, perform tests of that product, and demonstrate that product. For the systems engineering products these records are best retained in tools such as a requirements management tool, a model-based system engineering tool, and a configuration-controlled repository, to name a few. The development iteration is a short period of time, usually from two weeks to four weeks in duration.

3. For large products in development where multiple teams integrate their work items together to show a demonstratable product, several iterations may be needed to get to that point. This multiple iteration period is often referred to as an increment and frequently takes about three months.
4. Prior to starting an increment, all teams working to produce demonstrable products, should meet to plan their work, identify dependencies between the teams and establish commitments to meet the plan. This planning is called different things depending on the framework used by the program. Some call it the big room planning, some call it the Program Increment Planning.
5. At the end of the set of iterations the demonstrable product may be "released" to the stakeholders. Then all members of the program meet to plan the next increment of work.

A visual representation of teams working to this described cadence is shown in Figure 4 (Rosser et al. 2014).

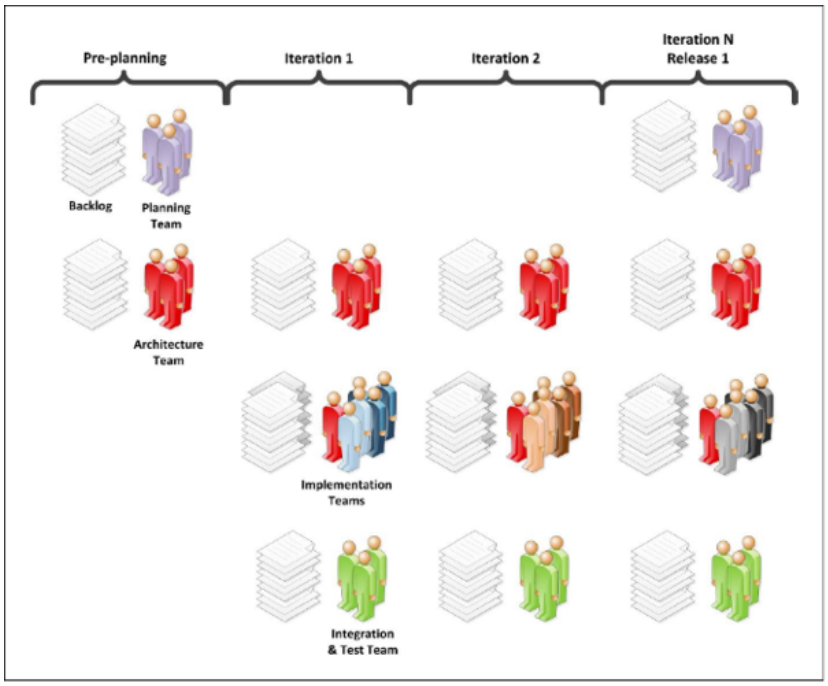

Figure 4. The Agile SE Framework (Rosser et al. 2014, Used with Permission)

This Agile Systems Engineering (SE) Framework (Rosser 2014) aligns with the Scaled Agile Framework (SAFe) depiction of teams working program and team backlogs using iterative development. SAFe is a framework that implements the principles of iterative development. It represents how a large system may have multiple life

cycle processes being followed in parallel over time. Key decision points need to be aligned between the multiple life cycle processes. There are many agile approaches that a program could use as is or combined to adapt to what works best for a given domain. The AgileAlliance (2017, 100) illustrates many of the “agile approaches based on their depth of guidance and breadth of their life cycles”.

For a complex system with changing requirements the assessment may result in the decision to use an incremental, iterative approach for development. Regardless of which model or framework is selected a program starts with a vision, a budget and usually a period of performance. Then the program’s stakeholders identify the highest value capability to develop first. The list of capabilities is prioritized so that the long-term development is visible. However, this prioritized order may change as work progresses. What is known about the intended product may have well defined requirements and architecture representations and what is conceptual will have those requirements and designs developed incrementally as time progresses. This incremental method of development is enabled by the use of an open system architecture, Model Based System Engineering (MBSE) tools, Set-based design, design thinking, continuous integration, continuous development, architecture patterns, microservice architecture, and Lean engineering.

References

Works Cited

Agile Alliance®, Project Management Institute (PMI). 2017. "Annex A3 Overview of Agile and Lean Frameworks," in *Agile Practice Guide*, Newtown Square, Pennsylvania: Project Management Institute, Inc. p. 100.

Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Greening, J., Highsmith, J., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D. 2001. Principles behind the Agile Manifesto. Accessed September 12, 2020. Available: Agilemanifesto.org/principles.html

Dove, R., Schindel, W. 2019. "Agile Systems Engineering Life Cycle Model for Mixed Discipline Engineering." Proceedings of the International Conference on Systems Engineering (INCOSE) International Symposium, July

20-25, 2019, Orlando, FL, USA.

Kusel, T. 2020. "A generic Systems Engineering process tailoring methodology, based on lessons from MeerKAT." Proceedings of the International Conference on Systems Engineering (INCOSE) International Symposium, July 20-22, 2020.

Marbach, P., Rosser, L., Osvalds, G., Lempia, D. 2015. "Principles for Agile Development." Proceedings of the International Conference on Systems Engineering (INCOSE) International Symposium, July 13-16, 2015, Seattle, WA, USA.

Rosser, L., Marbach, P., Osvalds, G., Lempia, D. 2014. "Systems Engineering for Software Intensive Programs using Agile." Proceedings of the International Conference on Systems Engineering (INCOSE) International Symposium, June 30-July 3, 2014, Las Vegas, NV, USA.

Scaled Agile Framework (SAFe®) Lean-Agile Principles. Accessed September 12, 2020. Available: https://www.scaledagileframework.com/safe-lean-agile-principles/

Schindel, W., Dove, R. 2016. "Introduction to the Agile Systems Engineering Life Cycle MBSE Pattern." Proceedings of the International Conference on Systems Engineering (INCOSE) International Symposium, July 18-21, 2016, Edinburgh, Scotland, UK.

Primary References

Agile Alliance®, Project Management Institute (PMI). 2017. "Annex A3 Overview of Agile and Lean Frameworks," in *Agile Practice Guide*, Newtown Square, Pennsylvania: Project Management Institute, Inc. p. 100.

Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Greening, J., Highsmith, J., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D. 2001. Principles behind the Agile Manifesto. Accessed September 12, 2020. Available: Agilemanifesto.org/principles.html

Dove, R., Schindel, W. 2019. "Agile Systems Engineering Life Cycle Model for Mixed Discipline Engineering." Proceedings of the International Conference on Systems Engineering (INCOSE) International Symposium, July 20-25, 2019, Orlando, FL, USA.

Kusel, T. 2020. "A generic Systems Engineering process tailoring methodology, based on lessons from MeerKAT." Proceedings of the International Conference on Systems Engineering (INCOSE) International Symposium, July 20-22, 2020.

Marbach, P., Rosser, L., Osvalds, G., Lempia, D. 2015. "Principles for Agile Development." Proceedings of the International Conference on Systems Engineering (INCOSE) International Symposium, July 13-16, 2015, Seattle, WA, USA.

Rosser, L., Marbach, P., Osvalds, G., Lempia, D. 2014. "Systems Engineering for Software Intensive Programs using Agile." Proceedings of the International Conference on Systems Engineering (INCOSE) International Symposium, June 30-July 3, 2014, Las Vegas, NV, USA.

Scaled Agile Framework (SAFe®) Lean-Agile Principles. Accessed September 12, 2020. Available: https://www.scaledagileframework.com/safe-lean-agile-principles/

Schindel, W., Dove, R. 2016. "Introduction to the Agile Systems Engineering Life Cycle MBSE Pattern." Proceedings of the International Conference on Systems Engineering (INCOSE) International Symposium, July 18-21, 2016, Edinburgh, Scotland, UK.

Additional References

Ward., D. 2014. *F.I.R.E. How Fast, Inexpensive, Restrained, and Elegant Methods Ignite Innovation*, New York, NY: Harper Collins Publishers.

< Previous Article | Parent Article | Next Article >
SEBoK v. 2.11, released 25 November 2024
