Requirements Engineering Innovations for Agile-Based Programs

Ms. Cynthia Harrison,
Dr. Barbara Pemberton
U.S. Army PEO STRI
Orlando, FL
cynthia.t.harrison.civ@mail.mil,
barbara.j.pemberton6.civ@mail.mil

Mr. Paul Butler, Mr. Bill Fetech,
Mr. Devin Hobby, Ms. Amy Lim
The MITRE Corporation
Orlando, FL
pvbutler@mitre.org, wfetech@mitre.org,
dhobby@mitre.org, ahblim@mitre.org

ABSTRACT

Since the Agile Manifesto was published in 2001, agile development practices have evolved in response to Department of Defense and industry frustration with long software development lead times and the inflexibility of technical and management decisions made early in the development lifecycle. With the increasing popularity and adoption of agile practices, organizations have focused primarily on transforming their approach to software development, less so on their practices in systems requirements engineering. Despite some progress and improvement in the application of agile development practices, agile techniques for requirements management remain a challenge because they have not matured to the level of the development practices. To operate effectively and efficiently, project teams must holistically adopt agile techniques throughout their organization and integrate them into all aspects of their requirements engineering and project management practices. This paper identifies and explores four significant challenge areas that affect innovation and adoption of agile solutions for requirements engineering: culture, requirements management, requirements definition, and tools as enablers. As part of this exploration, an integrated team of military domain experts (U.S. Army, U.S. Air Force) and agile systems engineering professionals from The MITRE Corporation analyzed each challenge area. These explorations utilized a case study approach to understand why and how innovation in agile system requirements engineering lags that of agile software development. Based on the case study data, the authors describe the impacts of cultural bias and how culture can change to improve organizational adoption of agile-based requirements engineering, agile techniques for managing requirements, contextual best practices for developing just-in-time requirements, and integrated tools that simplify time-intensive tasks associated with traditional requirements development.

ABOUT THE AUTHORS

Ms. Cynthia Harrison is the Technical Director (Acting) and Chief Software Architect at the U.S. Army Program Executive Office for Simulation, Training and Instrumentation (PEO STRI). Ms. Harrison is responsible for enterprise-wide system and software engineering standards, policy, and initiatives and Science and Technology planning. Ms. Harrison's experience includes acquisition and development of Live, Virtual, Constructive systems for training and simulation.

Dr. Barbara Pemberton is U.S. Army PEO STRI Lead for Systems Engineering Processes and Requirements Management. Dr. Pemberton's areas of expertise include requirements analysis, systems and software engineering, and acquisition and development of training systems for the Air Force, Navy, Army, and industry.

Mr. Bill Fetech is a Lead Multi-Discipline Systems Engineer with MITRE. He has 38 years of work experience and has authored over 15 MITRE Technical Reports on requirements management, developed and presented multiple classes on DOORS implementation, given presentations on requirements management at several conferences, and licensed two products to IBM. His current field of research is requirements management and modeling integration.

Mr. Paul Butler, Mr. Devin Hobby, and Ms. Amy Lim are MITRE systems engineers who support PEO STRI by designing and providing training for enterprise solutions that optimize technical and business processes. Their current field of research is agile requirements management; they are currently supporting the Army by developing a web-based training system that standardizes requirements processes, schemas, and tools for leaders and engineers.

Approved for Public Release; Distribution Unlimited. Public Release Case Number 19-2537
This technical data deliverable was developed using contract funds under Basic Contract No. W56KGU-18-D-0004.

© 2019 The MITRE Corporation.

Requirements Engineering Innovations for Agile-Based Programs

Ms. Cynthia Harrison,
Dr. Barbara Pemberton
U.S. Army PEO STRI
Orlando, FL
cynthia.t.harrison.civ@mail.mil,
barbara.j.pemberton6.civ@mail.mil

Mr. Paul Butler, Mr. Bill Fetech,
Mr. Devin Hobby, Ms. Amy Lim
The MITRE Corporation
Orlando, FL
pvbutler@mitre.org, wfetech@mitre.org,
dhobby@mitre.org, ahblim@mitre.org

INTRODUCTION

Army acquisition is becoming more agile. According to the *Army ALT Magazine* (Babbitt et al., 2019):

> The need is clear: Army acquisition must no longer be process-oriented, time-consuming and risk-averse, taking years to deliver a product. Enter the dynamic duo of middle-tier acquisition and other transaction authorities (OTA). Middle-tier acquisition is a mechanism to 'soft-start' a new program or rapidly advance an existing one by initiating a rapid prototyping effort. A program manager (PM) can acquire several prototypes for user assessment during the requirements development and maturation process, essentially allowing the users to 'buy, try and then decide.' But what about contracting, you ask? Certainly, if we speed up program timelines, traditional contracting will keep the overall process slow, right? Enter OTAs, which gives the contracting officer the capacity to enter into a legally binding contract directed to a specific vendor for the purchase of a prototype.

Many of today's Army agile acquisitions are implementing the acquisition approach described in the Middle Tier Section 804 of the National Defense Authorization Act acquisition (Interactive Adaptive Acquisition Framework Tool, n.d.). This approach supports rapid prototyping and rapid fielding of systems that need to be developed in less than five years. The acquisition is not subject to the Joint Capabilities Integration and Development System (JCIDS) Manual or Department of Defense Directive (DoDD) 5000.01. As a result, the acquisition documentation required is greatly reduced. Although this accelerated iterative middle-tier acquisition solves some existing problems, the Government would benefit from a more agile requirements management (RM) approach to further optimize the time it takes to deliver a product. Additional research is also needed to determine how to leverage agile processes and tools for acquisition.

To understand some of the history surrounding agile software development, flash back nearly two decades to 2001, when seventeen developers met in Snowbird, Utah, to discuss lightweight development methods. The outcome of their meeting was the Manifesto for Agile Software Development, which defined four agile core values (Beedle et al., 2001):

- Focus more on individuals and interactions than on processes and tools.
- Working software is more important than comprehensive documentation.
- Customer collaboration is more important than contract negotiation.
- The process should respond to change rather than follow a plan.

A key tenet of this agile philosophy is to always accept changes in requirements no matter how early or late in the development cycle. As Government acquisition becomes more agile, the RM process must become more agile as well, in all phases of the acquisition cycle.

The need for applying agile principles and values is well documented (Abrahamsson et al., 2017; Bjarnason et al., 2017; Kraus, 2017). There is general agreement that these tenets increase the likelihood of a product that meet user needs:

- Ability to adapt to changing requirements throughout a project / development process
- Increase speed-to-market
- Improve product and process quality
- Increase transparency and visibility into progress of projects and development of products

According to *The Requirements Engineering Handbook* (Young, 2004), "Errors that originate in requirements tend to be the most expensive and troublesome to eliminate later" and "more than half of all defects can be traced to requirements errors." The need for agile practices becomes evident through problems that arise in traditional project teams when all of the requirements must be defined up front. There is an implicit trust that systems engineers (designers and developers) will correctly interpret and implement the requirements. But, as with any engineering product, requirements specifications may contain errors, may not be sufficient, and may change. When available, engineers consult with end-users for more information; if not, they interpret the meaning and intent of requirements which can lead to defects that are not discovered until much later.

Agile project teams recognize that requirements are not always well understood up front. They develop user stories to capture and elicit system or software features from an end-user perspective. The Army PEO STRI Persistent Cyber Training Environment (PCTE) program is one example of an agile systems engineering success story. In less than one year after having received its funding, PCTE released two cyber training prototypes. The Army attributes this success to its *Try-Buy-Adapt* agile acquisition strategy (an adaptation of the 'buy, try, and then decide') and key agile practices like use of prototyping and frequent end-user engagement to drive the requirements process (Kapadia & Osborne, 2019).

Centered around the following Army PEO STRI Case Study, this paper focuses on four challenge areas for innovation and adoption of agile solutions for systems and requirements engineering:

- Cultural bias against agile adoption, and how culture can change to improve organizational adoption of agile-based RM
- Agile techniques for RM
- Contextual best practices for developing just-in-time requirements
- Integrated tools that simplify time-intensive tasks associated with traditional requirements development.

The paper discusses the four challenge areas in the context of the case study and derives conclusions and recommendations.

U.S. ARMY CASE STUDY

For several years, multiple U.S. Air Force (AF) and U.S. Army organizations have been developing and implementing RM processes and tools. The AF has had great success implementing processes for requirements engineering on a number of programs. These processes and tools are being reviewed and implemented by PEO STRI to contribute to the development of their own processes. In most of those instances, after evaluation of multiple RM tools, the Dynamic Object Oriented Requirements System (DOORS) was identified as the best tool (Fetech, 2010; Gorski et al., 2013). In 2017, the U.S. Army Program Executive Office for Simulation, Training, and Instrumentation (PEO STRI) established a policy to implement DOORS across the enterprise. The PEO STRI enterprise consists of PMs operating with autonomy to deliver complex solutions within compressed schedules. The PEO's objective for technical and acquisition excellence across the enterprise is to focus on the core capability areas of requirements engineering.

As a first step, the organization created a Requirements Management Working Group (RMWG), chartered to assess, design, and institutionalize a common technical requirements approach across the PEO. The technical requirements approach would not constrain projects to one methodology over another (e.g., agile, traditional, product line), but achieve efficiencies and effectiveness through strategic reuse of shared assets and processes. The RMWG comprises Army acquisition subject matter experts and engineers and systems engineers from The MITRE Corporation. The RMWG reports progress monthly to a Council of Chief Engineers who in turn report directly to the Technical Director of PEO STRI.

Initially, the RMWG developed a "campaign" plan to assess needs across the enterprise by developing a vision and engagement strategy with PEO leadership and the PMs. The RMWG then developed a roadmap: a schedule of the enterprise requirements goals, products, and events on a timeline. The RMWG also designed a strategic communications strategy that described and advocated the requirements efforts (with a focus on agile, but also including traditional and product line approaches) outlined in the roadmap. These steps contributed to developing the culture necessary to apply agile practices in systems requirements engineering.

Finally, the RMWG coordinated with PMs to pilot requirements schemas, processes, and tools (described in the Requirements Structure section). The first pilot was conducted with a PM who was in post-contract award and employed the traditional methodology for requirements development and received contractor-expanded requirements in DOORS files. The RMWG is applying lessons learned and best practices from the initial pilot to subsequent pilots based on OTAs and employing a Scaled Agile Framework® (SAFe®) methodology (described in the SAFe® Tailoring Recommendations and Best Practices section).

The following sections illustrate AF exemplar processes and tools that the RMWG is using for Army PEO STRI follow-on pilots. Also included are lessons and practices that the RMWG has applied from its examination of AF programs that have undergone similar transitions.

CULTURAL BIAS AGAINST AGILE ADOPTION

Agile is often thought of as a set of processes, procedures, tools, databases, and assets; actually, it is more of a culture. Culture is defined as the collective norms, rituals, and behaviors that a group of people share, or, more colloquially, as the 'way of life' (*Cambridge English Dictionary,* 2019) of people, meaning the way they do things. Culture encompasses the values, beliefs, systems of language and communication, and practices that members of an organization have in common and that can be used to define them as a collective entity. An agile culture is one that has adopted and implemented the core values and principles of the Agile Manifesto in its day-to-day operations: specifically, one where iterative development methodologies are employed and where requirements and solutions evolve through collaboration among self-organizing cross-functional teams.

Often compared to and contrasted with agile is the waterfall (or traditional) software development methodology, which employs a linear approach to the work of software development. A typical waterfall sequence comprises (1) defining / documenting system requirements; (2) designing the architecture; (3) implementing the design, performing testing and integration, and obtaining user acceptance; (4) delivering / fielding the system; and (5) sustaining and maintaining the system. The waterfall process can produce high-quality products and prove advantageous in certain situations (Davis, 2013). For example:

- Projects that are small in size.
- Projects that are simple in complexity.
- Projects that incorporate many hardware components that would be expensive to adapt.
- Projects that incorporate safety systems that require full functionality before being released.

By contrast, agile is an iterative approach that emphasizes rapid delivery of user-prioritized capabilities by incorporating all phases of the software lifecycle in a continuous cycle. Both agile and waterfall have their positive and negative aspects, with agile in most recent years gaining acceptance by industry, government, military, and research organizations. Agile has the following clear advantages over the waterfall method:

- End users are involved throughout the entire development process, from requirements definition through delivery, enabling the developer to understand and focus on delivering the key capabilities earlier in the process.
- Agile supports late changes to requirements.
- A product owner or customer / user representative has more frequent opportunities to refine or reprioritize features and capabilities of the product.
- Agile enables continuous delivery of the working product.

Employing agile practices, PCTE tries, buys, and adapts commercial capabilities through competitive Cyber Innovation Challenges (CICs), enabling operational users to provide early and frequent feedback to developers on feature and capability priorities (Kapadia & Osborne, 2019).

Barriers

Significant barriers remain for organizations that are considering adopting agile, whether incrementally or across all their development teams. The following list identifies several significant barriers to adopting and developing agile requirements processes, tools, and mindsets (Huether, 2013; Cottmeyer, 2011):

- Lack of support from leadership, particularly at the highest levels
- Organization philosophy and culture that conflict with agile's core values and principles

- Lack of understanding and experience with agile methodologies
- Lack of communication, vertically and horizontally, to promulgate an understanding of the shortcomings of traditional methodologies (e.g., waterfall) and agile process improvements
- Pressure from within to maintain "laissez-faire" and continue development with traditional methodologies, such as waterfall
- Lack of training and guidance for not just engineering staff but all stakeholders, including leadership / decision makers.

All of these barriers are related to culture, either directly or indirectly. Organizations must focus on establishing an agile mindset to achieve a cultural shift.

Cultural Bias Recommended Practices

To address some of these challenges and to overcome these barriers, PEO STRI has adopted a culture that supports planning for and developing agile practices in systems requirements engineering. This required the organization to implement the following essential steps:[1]

1) <u>Engage with the organization's leadership</u>, to include business and information technology (IT) leaders/managers, to shape policy for agile adoption. This step should be familiar to business professionals, as both vertical and horizontal engagement involving advocacy as well as inquiry is a key principle of strategic leadership. Such engagement not only informs leaders, but also motivates them to assume ownership, responsibility, and authority. Engagement with leadership must occur at all levels: "[w]e hold all leaders accountable, to protect our investment and agile commitment" (Rometty, 2018). At PEO STRI, the RMWG engaged with leadership, including the Technical Director and Chief Engineers, who directed the PMs to implement PEO enterprise RM standards. When opportunities are presented to and embraced by the business and technology leadership of an organization, leaders begin to own the technical baseline, making informed and timely decisions to manage risk, cost, and schedule while enacting business and process policies that ensure efficient and effective project execution.

2) <u>Develop a transition roadmap</u> from current practices to agile. It is not likely (nor would it be advisable) that an organization with responsibility for a few to many programs would transition to agile all at once. As with any business / IT decision, the path to agile must be carefully planned and prepared. Prior to the formation of the RMWG, a PEO STRI investigatory team defined an agile transition roadmap prioritized by impact on "owning the technical baseline."

Education and training are essential first steps in transitioning from traditional systems engineering methodologies. Several AF cryptologic programs (Defense Systems, n.d.; Mini-crypto, n.d.; Nair, 2017; Viasat Receives Initial Mini Crypto Order from the U.S. Air Force Launching Production, n.d.) have successfully implemented the education, training, and requirements engineering approaches now being implemented by the RMWG at PEO STRI. Because of this education and training these programs were able to adopt new development strategies by linking the agile concept to the organization's vision, thereby implementing standards and requirements engineering processes across their acquisition portfolio. Thereafter, multiple AF cryptologic programs adopted Model Based Systems Engineering (MBSE) techniques in lieu of a document-centric approach. Each program has successfully completed multiple technical reviews by identifying requirements issues, such as orphan and widow requirements, earlier in the lifecycle. One AF cryptologic program (Newell, 2017) has recently entered Milestone C and is beginning full-rate production. In the Army's initial pilot (see the Army Case Study), the RMWG demonstrated this transition by leveraging AF lessons learned and best practices to help deliver the message and illustrate how the transition can be successful by using many of the same MBSE approaches developed by the AF.

A second step in an effective transition is communication and collaboration. Programs must develop and follow a "top down" communication strategy to present the need and scope of the transition and how teams will be affected and expected to support the strategy. This provides transparency, and clearly and consistently articulates to the entire

[1] These steps have been formulated from years of combined author experience and expertise developed by working across the DoD, with a focus on Army and AF programs. These practices could be adapted for other non-traditional approaches as well and implemented with successful outcomes.

organization understanding and support for the transition. At PEO STRI, the RMWG developed an "elevator speech" to address cultural bias and address perceived roadblocks when using DOORS and transitioning to agile, and plans to develop a "TED-like" talk as well.

3) Embolden and empower individuals and teams. "People and culture are superpowers – empower and coach" is cited as the key reason for the German Daimler AG carmaker's unparalleled agility, performance, and relevance in the 21st century (Neumann, 2018). To support a broader cultural adoption of agile, the PEO STRI RMWG encouraged teams to maintain flexibility in executing their specific functions, including:

- Holding conversations with end users to understand required capabilities
- Following practices and leveraging tools used to manage information obtained through end-user coordination and collaboration
- Supporting automated and continuous integration and testing of software / solutions to ensure capabilities are properly implemented
- Striving to provide near-continuous product availability to end users for their feedback.

4) Ensure that individuals and teams remain flexible. Flexibility is intrinsic in agile ("Welcome changing requirements, even late in development") (Beedle et al., 2001). End users and customers change their minds, especially as they see their product beginning to evolve. The organization and its management must promote the philosophy that changes improve a product, rather than subvert deliveries. Prior to the advent of agile and even currently, PEO STRI manages ever-changing requirements through Technical Refresh, a methodology for enabling program and organizational flexibility by managing not only requirements dynamics, but that of the total program life cycle.

5) Perform pilots with a trusted, experienced agile partner / guide. Adopting / developing / tailoring / evolving agile requirements practices for projects requires strong support from leadership and management. The experienced guide can help the organization avoid typical pitfalls and make informed decisions, leveraging his or her experience, lessons learned, and best practices. The guide helps the organization focus not only on process and technology but also on cultural development. The first pilot serves as a catalyst for follow-up pilots, providing guidance and clarity to the complexities, risks, and cost (in terms of technology, labor, and schedule). As mentioned in the PEO STRI Case Study, the RMWG is making extensive use of pilots to train engineering staff and leadership. Metrics are currently being collected and analyzed, and early user feedback indicates that a cultural adoption of agile practices is underway.

AGILE TECHNIQUES FOR REQUIREMENTS MANAGEMENT

The process to specify and decompose user requirements into a format needed for agile software development is not well defined. In this section, we show how this challenge can be addressed by leveraging agile software development methodologies such as Extreme Programming (XP) (Extreme Programming: a gentle introduction, 2013); Scrum (What is SCRUM?, n.d.); or Kanban (Kanban, n.d.). SAFe® (Portfolio SAFe®, 2018) provides the techniques and processes to implement an agile systems requirements engineering approach while integrating with methodologies such as Lean and DevOps. The AF and Army are just now beginning to investigate these methodologies.

Requirements Structure

Figure 1 depicts the SAFe® four-tier hierarchy and process flow (Portfolio SAFe®, 2018). Each tier addresses a portion of the requirements methodology. In the Portfolio tier, the requirements are defined as Epics. An Epic is "a container for a Solution development initiative large enough to require analysis, the definition of a Minimum Viable Product (MVP), and financial approval before implementation" (SAFe® Glossary, 2019). The second tier, Large Solution, which is optional for smaller solutions, derives Capabilities from the Epics. Capabilities are defined as "a higher-level solution behavior that typically spans multiple Agile Release Teams. They are sized and split into multiple Features to facilitate their implementation in a single Program Increment" (SAFe® Glossary, 2019). The next tier is the Program tier, where the Capabilities (if the Large Solution tier is used) or the Epics are decomposed into Features. A Feature is defined as "a service that fulfills a stakeholder need and is sized to be delivered by a single Agile Release Train in a Program Increment" (SAFe® Glossary, 2019). The lowest tier, Team, is where the Features are decomposed into Stories. Stories are defined as "short descriptions of a small piece of desired functionality, written in the user's language. They are sized to be implemented in small, vertical slices within a single Iteration" (SAFe® Glossary, 2019).

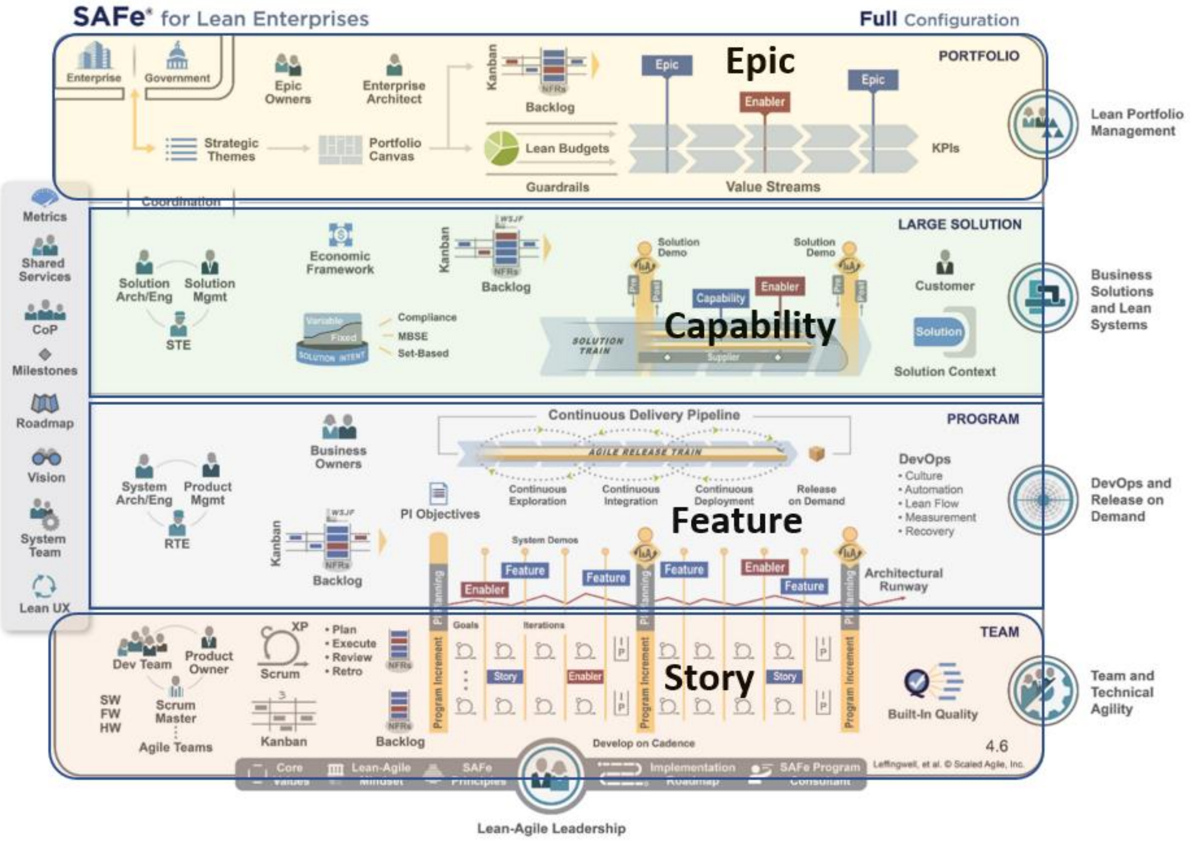


Figure 1. SAFe® Large Solution Configuration

Requirements Data Captured at Each Tier

The four SAFe® tiers identified above provide a scalable requirements model that depicts a way to express the system behaviors of Epics, Capabilities, Features, and Stories. When defining the Capabilities, Features, and Stories, some of the same Epic attribute names are used, but each level adds unique attributes. Table 1 contains the Epic attributes (IBM Program Epic, 2018) that IBM tailored from the set of attributes defined in SAFe®. The RMWG is further tailoring this set of attributes as part of PEO STRI's enterprise standardization. This paper does not cover the attributes needed for the other tiers (Capabilities, Features, Stories).

Table 1. Agile Requirements Management Recommended Epic Attributes

Attribute Name	Data Type / Values	Description
Estimated Story Points	Integer	Integer representing the level of effort expected to complete a work item[2]
Actual Story Points	Integer	Determined and entered after Epic is implemented
Epic Hypothesis Statement	Object Text	Objective
Epic Funnel Status	List: *Draft*, Final	Attributes for maintaining the current status of the Epic.
Epic Review Status	List: *Awaiting Review*, Reviewing	

[2] In agile, a *work item* is the term used to track different types of work, such as user stories, bugs, and issues.

Attribute Name	Data Type / Values	Description
Epic Analysis Status	List: *Awaiting Documentation*, Analyzing, Ready for Approval, Approved	Note: Italicized values are the default values that are set when the Epic is created.
Epic Closed Status	List: *Open*, Accepted, Rejected	
Proposed Iteration	Text	Target iteration on the roadmap for this work item
SAFe® Enabler	List: *Architecture*, Compliance, Exploration or Infrastructure	Used to categorize the work item as work required for specified area
SAFe® Work	List: *Business* or Enabler	Defines the work type as Business or Enabler
Success Criteria	String	How success will be determined upon delivery of this work item
Weighted Shortest Job First (WSJF)	Calculated value http://www.scaledagileframework.com/wsjf	Uses Job Size, User or Business Value, Time Criticality, and Risk Reduction Opportunity Enablement Value (RROEV), not included in this table.

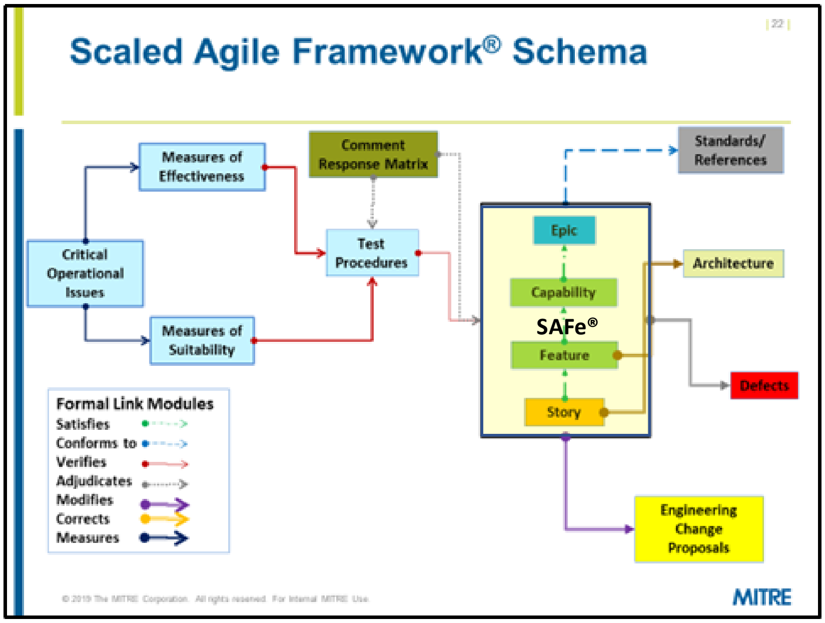
SAFe® Tailoring Recommendations and Best Practices

The PEO STRI RMWG conducted an analysis of the full SAFe® methodology. This analysis consisted of reviewing the SAFe® hierarchy, forms used by the SAFe® methodology, and attributes defined at each of the SAFe® tiers. Based on this analysis, the PEO STRI RMWG identified and documented areas where the SAFe® methodology could be tailored to meet the needs of PEO STRI for developing requirements. In future Army pilots, the RMWG will use this tailoring and the best practices listed in Table 2.

Table 2. PEO STRI Tailoring Recommendations and Best Practices for SAFe®

Analysis Area	Tailoring Recommendations and Best Practices
SAFe® Hierarchy	- If the four-tier hierarchy is implemented, the "Large Solution" must be documented - Most organizations implementing SAFe® need only the three-tier hierarchy
SAFe® Forms	- Define the hypothesis statement in a single text attribute as specified in Table 1
SAFe® Attributes	- For each SAFe® tier implemented, review the attributes and tailor them to the needs of the organization - Table 1 is a tailored version of the full SAFe® Epic-level attributes
Training	- Obtain training on SAFe®
Metrics	- Identify metrics to capture - Identify how each metric will be captured and used - Identify the organization that will capture and maintain each metric - Identify tools needed to capture each metric

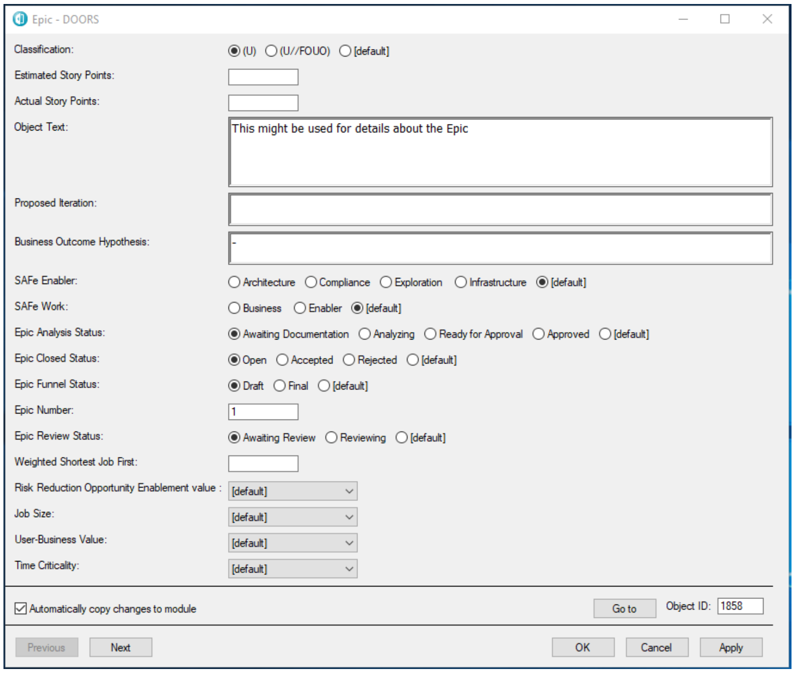
BEST PRACTICES FOR JUST-IN-TIME REQUIREMENTS



Scaled Agile Framework® Schema

Figure 2. DOORS SAFe® Schema

SAFe® is just one aspect of implementing MBSE. However, organizations using MBSE techniques to support just-in-time requirements development face challenges resulting from (1) inconsistent utilization of tools to support the requirements process and (2) current practices that focus on complete requirements decomposition. To address the former, several AF and Army programs have implemented DoD RM frameworks in DOORS to support the Joint Capabilities Integrated Development System (JCIDS) (Fetech, 2010; Fetech, 2016a; Fetech, 2016b; Fetech, 2016c; Fetech & Shaffer, 2016; Gorski et al., 2013; Gorski, Fetech, 2013). To address the latter challenge, the PEO STRI RMWG is using the JCIDS Weapon Systems framework as the basis for the DOORS SAFe® schema depicted in Figure 2, which also highlights the four SAFe® tiers described in the previous section. The DOORS SAFe® schema integrates the four tiers with other RM features (standards/references, defects, and engineering change proposals) and systems engineering processes (architecture and testing). Using the DOORS SAFe® schema, the RMWG utilized the capabilities of DOORS to rapidly capture the data needed at each tier. This involved the use of DOORS forms to quickly gather and analyze the attributes identified for each tier. Figure 3 depicts an RMWG-developed DOORS form for the SAFe® Epic tier attributes.



Epic - DOORS

Classification:	● (U) ○ (U//FOUO) ○ [default]
Estimated Story Points:	
Actual Story Points:	
Object Text:	This might be used for details about the Epic
Proposed Iteration:	
Business Outcome Hypothesis:	-
SAFe Enabler:	○ Architecture ○ Compliance ○ Exploration ○ Infrastructure ● [default]
SAFe Work:	○ Business ○ Enabler ● [default]
Epic Analysis Status:	● Awaiting Documentation ○ Analyzing ○ Ready for Approval ○ Approved ○ [default]
Epic Closed Status:	● Open ○ Accepted ○ Rejected ○ [default]
Epic Funnel Status:	● Draft ○ Final ○ [default]
Epic Number:	1
Epic Review Status:	● Awaiting Review ○ Reviewing ○ [default]
Weighted Shortest Job First:	
Risk Reduction Opportunity Enablement value :	[default]
Job Size:	[default]
User-Business Value:	[default]
Time Criticality:	[default]

☑ Automatically copy changes to module

Object ID: 1858

Figure 3. Epic Data Entry / Update Form

Just-In-Time Requirements Recommendations

Subsequent PEO STRI pilots will use RMWG-designed forms for the other tiers to support the agile requirements development approach. The RMWG also automated the linking process that enhances the agile approach by abstracting the structure of the DOORS schema employed by the user.

The SAFe® methodology integrates with the user requirements process at the Epic level and with the agile software engineering practices at the Story level. The end goal of implementing the SAFe® methodology is to have sufficient Stories (probably a maximum of three sprints) queued-up for the agile development teams so they always have Stories to implement. The RMWG recommends that programs or enterprises using or planning to use the SAFe® methodology:

- Understand the reason the program or enterprise is moving toward agile
- Obtain participation by individuals who have used the SAFe® methodology
- Use the DOORS schema discussed in this document and tailor it to fit program / enterprise needs
- Develop, distribute, and implement a policy enforcing the use of the above practices.

INTEGRATED TOOLS FOR AGILE REQUIREMENTS DEVELOPMENT

RM tools mitigate the burdens of time-intensive tasks by enabling users to effectively capture, trace, analyze, and manage changes to information throughout the development lifecycle. Requirements engineers (REs), leveraging their training in agile methodologies, use RM tools to develop specific, measurable, time-bound requirements. Agile software engineers extract and decompose these requirements into clear, concise, prioritized tasks for design and implementation. With the assistance of RM tools, organizations are thus able to create just-in-time requirements and derive tasks that adapt to project needs, resulting in rapid development cycles. Depending upon the project and other considerations, organizations may choose to acquire commercial RM tools or develop customized solutions and aids that help engineering staff effectively manage the project. These aids serve as organizational standards by providing a common set of reusable management and technical processes, schemas, training, wizards, and guides.

Requirements Management Tools Assessment

In order to adopt agile processes, organizations must have the right set of management tools. However, determining the proper RM tool is often a challenge due to the vast number of solutions available. These tools can range from being very basic to extremely complex in terms of usability and function. The tools listed in Table 3 are a few examples of RM tools our team evaluated based on consumer popularity and their diverse set of capabilities.

Table 3. Commercial RM Tools

Tool	Developer	Description
DOORS	IBM	DOORS is an RM tool that captures, traces, analyzes, and manages changes to information. DOORS is customizable, and its main distinguishing feature is its interoperability with other tools such as MagicDraw.
Modern Requirements	Modern Requirements	Modern Requirements provides a comprehensive and easy-to-use RM Suite (Trace Analysis, Impact Analyzer, Reporting, etc.) that is built into Microsoft (MS) Azure DevOps and can be implemented on-premises or in the cloud.
Visure	Visure Solutions	Visure provides a comprehensive, collaborative Application Lifecycle Management (ALM) platform. Features include full traceability, integration with MS Word and Excel, risk management, test management, and more.
ReQtest	ReQtest	ReQtest is a cloud-based requirements, test management, and bug tracking tool with a built-in agile board to help organize, prioritize, and delegate tasks with ease.
JIRA & Confluence	Atlassian	JIRA can be used with Confluence for RM purposes. The Atlassian Marketplace also has many applications available for RM that can be customized and tailored to fit a program's needs.

Tool	Developer	Description
Dimensions RM	Micro Focus	Dimensions RM is a comprehensive web-based RM tool. It supports both agile framework and traditional methodologies. Agile and traditional artifacts can be viewed and linked together to support hybrid methodologies.

RM tools should help an organization standardize, enforce, and formalize a common set of requirements specifications throughout the development lifecycle. Most tools provide those capabilities for an organization. To select an appropriate tool, the organization should first determine what needs a tool must meet. The following are criteria that should be used to evaluate RM tools:

- Overall cost vs. allotted budget
- Cloud-based or on-premises
- Built-in capabilities such as traceability, impact analysis, scalability, and collaboration
- Built-in modeling tool or capability for integration with modeling tools (e.g., No Magic MagicDraw, IBM Rhapsody, Sparx Enterprise Architect, Vitech CORE, and UNICOM SI System Architect)
- Integration with other applications (e.g., MS Word, Excel, JIRA, Git, Agilefant, or other RM tools)
- Additional features such as bug tracking, risk management, and test management.

Table 4 presents a summary of the commercial RM tools in Table 3, evaluated based on the above-listed criteria. Cost is omitted as part of the assessment since it will vary depending on each organization's circumstances.

Table 4. Criteria Assessment of RM Tools

Tool Name	Cloud Platform	Built-in Traceability and Impact Analysis	Integration with Modeling Tools or Built-in	Integration with Other Applications	Additional Built-in Features
DOORS	✓	✓	✓	✓	✓
Modern Requirements	✓	✓	✓		
Visure	✓	✓	✓	✓	✓
ReQtest	✓	✓		✓	✓
JIRA & Confluence	✓		✓	✓	✓
Dimensions RM		✓	✓	✓	✓

Developed Guides

Commercial tools are helpful in the overall RM process, and understanding how to use and tailor each tool to the specific environment and acquisition strategy presents a considerable challenge. Training on RM tools provided by commercial corporations may not be comprehensive enough to enable optimum use of the tools for a specific need or project. Since PEO STRI has mandated the use of DOORS across the organization, they have established standardized schemas and best practices to ease implementation concerns. In addition, The MITRE Corporation developed a tool called the DOORS Help and Instruction Guide (HInGe), a user-friendly front-end web application designed for REs to use in each step of their RM process. HInGe contains schemas, training materials, and an easy-to-use step-by-step wizard. It does not necessitate any in-depth knowledge of the RM process or DOORS; instead, it enables novice and expert REs alike to effectively capture, trace, analyze, and manage changes across the development cycle.

Tool Support for Agile Teams Recommendations

"It's integrating different stakeholders and getting everyone on a common platform that ultimately provides traceability, and bridges the gaps between [company] teams, its customers and partners."

- Naresh Choudhary, Vice President, Infosys

In 2010 (Fetech, 2010), the AF developed a basic requirements schema for its cryptologic programs and implemented it in DOORS. The Army (Gorski & Fetech, 2013) and AF (Fetech, 2016b; Fetech, 2016c) used that schema as the basis for their DOORS implementations. During the process of tailoring the AF requirements schema and acquiring/developing RM tools, the RMWG documented lessons learned and best practice recommendations concerning tool support when using agile methodologies.

Any tool can be agile. Agile is a workflow architecture, not a system or tool. Some tools are set up to follow common agile practices, but the organization is responsible for establishing and executing agile methodologies.

- Organizations should establish a baseline of products to ensure consistent and effective results. Create and integrate templates and schemas into the tools to increase efficiency and decrease errors.
- Organizations should make the tool fit the team, not the team fit the tool. Customize settings and write scripts to complete tasks quickly and correctly. Ask: How will the tool enable REs to accomplish their tasks? How will it integrate with other tools used by the team? How long will it take to train the team on its use?
- Teams should reach out to other teams that have similar projects and use the same tools to understand any pitfalls they encountered along the way. Use templates, scripts, schemas, and other aids already generated by others.
- Many tools automatically keep track of changes, display differences, maintain legacy versions, and have server-based content management systems. Always keep requirements organized and record all changes.
- Most requirements tools include an impact analyzer to drive decision making. Before making changes to the requirements, use the impact analyzer to understand the effect a change will have on the effort. The result may cause the team to reconsider making the change, postpone the change until a later date, or change the scope of the current effort.
- Training provides knowledge and understanding of a tool's capabilities. Generate additional training tailored to the organization to teach REs how to leverage those capabilities.

CONCLUSION

As evidenced in the PEO STRI Case Study, overcoming cultural bias can present a significant challenge to organizations intending to develop and adopt agile-based requirements processes and tools. This bias is typically the result of lack of organizational support, lack of understanding or experience, and/or insufficient communication (whether with leadership or at the technical level). However, a cultural shift can be achieved by engaging early and continuously with an organization's leadership; developing an executable transition strategy from current practices to agile; empowering individuals and teams to take ownership of their planning, preparation, and technical processes; and teaming with an experienced partner or guide to execute pilots fostering incremental adoption of agile.

AF cryptologic programs have successfully proven that moving to MBSE approaches provides significant increases in requirements stability, thereby reducing cost and schedule while producing a system that satisfies user needs. By implementing an agile requirements development approach based on methodologies like SAFe® and best practices for just-in-time requirements, programs can achieve improvements similar to those experienced by AF cryptologic programs.

To adopt agile processes, organizations must have the right set of RM tools. Tools should help the organization standardize, enforce, and formalize a common set of requirements specifications throughout the development lifecycle. As noted above, organizations should make the tool fit the team, not the team fit the tool.

To paraphrase *Army ALT Magazine*, the need is clear, and the time is now. Agile is mainstream: in common use for 20 years. Countless organizations are benefitting from agile in both engineering and acquisition. Challenges remain; however, these challenges can be managed and overcome by following recommendations from experts. Organizational demand for efficiency, creativity, and innovation (Mini Crypto (MC) Revolutionizing Crypto for Tactical Ops., 2018) has increased and will not subside. Agile presents a methodology that is well aligned and mature enough to deliver products that meet user needs.

REFERENCES

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile Software Development Methods: Review and Analysis. Espoo: Technical Research Centre of Finland.

Aston, B. (2019, April 11). The Best Requirements Management Tools of 2019. Retrieved April 23, 2019, from https://thedigitalprojectmanager.com/requirements-management-tools/

Atlassian. (n.d.). Software Development and Collaboration Tools. Retrieved April 23, 2019, from https://www.atlassian.com/

Babbitt, J.D., & Schlomer, D. (2019, April 22). The Need for Speed. *Army AL&T Magazine Spring 2019*. Retrieved April 26, 2019, from https://asc.army.mil/web/news-alt-amj19-the-need-for-speed/

Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Hunt, A., Jeffries, R., Kern, J., Marick, B., Schwaber, K., Sutherland, J., & Thomas, D. (2001). Manifesto for Agile Software Development. Retrieved April 29, 2019, from https://agilemanifesto.org.

Bjarnason, E., Wnuk, K., & Regnell, B. (2017). A case study on benefits and side-effects of agile practices in large-scale requirements engineering. Lancaster University, UK: ACM.

Cambridge English Dictionary (2019). Cambridge, UK: Cambridge University Press.

Cottmeyer, M. (2011). 12 Key Barriers to Agile Transformation. Retrieved May 16, 2019, from https://dzone.com/articles/12-key-barriers-agile

Davis, B. (2013). Agile Practices for Waterfall Projects: Shifting Processes for Competitive Advantage. Plantation, FL: J. Ross Pub.

Defense Systems. (n.d.). Air Force wants 'mini crypto' devices for battlefield ops. Retrieved May 15, 2019, from https://defensesystems.com/articles/2014/04/17/air-force-mini-crypto-program.aspx

Extreme Programming: A gentle introduction (2013). Retrieved May 6, 2019, from http://www.extremeprogramming.org/

Fetech, B. (2010). Implementation of IBM’s Dynamic Object Oriented Requirements System (DOORS) at Cryptologic Systems Group (CPSG), MITRE Technical Report (MTR) 160064.

Fetech, B. (2016a) Implementing IBM Dynamic Object Oriented Requirements System Partitioning at Cryptologic and Cyber Systems Division, MTR 100065.

Fetech, B. (2016b). Owning the Technical Requirements Baseline, MTR 160144.

Fetech, B. (2016c). Owning the Technical Requirements Baseline – Integrating Multiple External Developers, MTR 170004.

Fetech, B., & Shaffer, D. (2016). Using DOORS for Requirements Contract Deliverables, MTR 160466.

Gorski, B., & Fetech, B. (2013). Common Operating Environment Integrated Requirements Framework, MTR 130057.

Gorski, B., Fetech, B., Chadwick, B., & Farrell, D. (2013). Mission Command Requirements Governance Team Requirements Tool Kit Analysis, MTR 120613.

Huether, D (2013). Barriers to Agile Adoption. Retrieved May 16, 2019, from https://www.leadingagile.com/2013/11/barriers-to-agile-adoption/

IBM Program Epic (2018). Retrieved April 17, 2019, https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/W54ecb028c53d_48b0_9d5e_4584a00489d3/page/Program%20Epic

IBM Rational DOORS Family. (n.d.). Retrieved April 23, 2019, from https://www.ibm.com/us-en/marketplace/rational-doors

Interactive Adaptive Acquisition Framework Tool. (n.d.). Retrieved August 8, 2019, from https://www.dau.mil/tools/t/Interactive-Adaptive-Acquisition-Framework-Tool

Kanban. (n.d.), Retrieved May 6, 2019, from https://www.agilealliance.org/glossary/kanban/

Kapadia, A., & Osborne, R. (2019) TRY-BUY-ADAPT, NATO Modelling and Simulation Group MSG-175 16th Workshop, Paris, 26 September 2019. Washington: The MITRE Corporation.

Kraus, A. (2017). Agile Explained: An Overview of the Values and Benefits of Agile. SC: CreateSpace Independent Publishing

Mini-crypto. (n.d.). Retrieved May 27, 2019, from https://www.wpafb.af.mil/News/Photos/igphoto/2001818625/

Mini Crypto (MC) Revolutionizing Crypto for Tactical Ops. (2018). U_Mini Crypto Overview (FVEY Releaseable) 30Apr18 [PDF file]. Internal report Air Force Life Cycle Management Center/Cryptologic and Cyber Systems Division): unpublished.

Nair, S. (2017, October 25). USAF develops new Mini Crypto chip to secure communications. Retrieved May 15, 2019, from https://www.airforce-technology.com/news/newsusaf-develops-new-mini-crypto-chip-secure-communications-5941318/

Newell, B. (2017). A Chip that Scrambles. Retrieved May 17, 2019, from https://www.af.mil/News/Article-Display/Article/1331042/a-chip-that-scrambles/

Neumann, U. (2018, October 17). Innovation and Transformation: The New IT, CIO Story Session Keynote Speaker, CIO Daimler Financial Services, Gartner IT Symposium/Xpo, Orlando, FL.

Portfolio SAFe®. (2018). SAFe® for Lean Enterprises, Retrieved April 18, 2019, from https://www.scaledagileframework.com/#

Requirement Management Tools and Software. (n.d.). Retrieved April 23, 2019, from https://www.modernrequirements.com/

Requirements Management Tool. (n.d.). Retrieved April 23, 2019, from https://visuresolutions.com/

Requirements, Test Management, Bug Tracking Tool. (n.d.). Retrieved April 23, 2019, from https://reqtest.com/

Rometty, G. (2018, October 16). Guest Keynote: Mastermind Interview with Chairman, President, and CEO of IBM, Gartner IT Symposium/Xpo, Orlando, FL.

SAFe® Glossary (2019). Retrieved April 18, 2019, from https://www.scaledagileframework.com/glossary/

Top 20 Best Requirements Management Tools (The Complete List). (2019, April 23). Retrieved April 23, 2019, from https://www.softwaretestinghelp.com/requirements-management-tools/

Viasat Receives Initial Mini Crypto Order from the U.S. Air Force Launching Production. (n.d.). Retrieved May 15, 2019, from https://www.viasat.com/news/viasat-receives-initial-mini-crypto-order-us-air-force-launching-production

What is SCRUM? (n.d.) Retrieved May 6, 2019, from https://www.scrum.org/resources/what-is-scrum

Young, R. R. (2004). The requirements engineering handbook. Boston, MA: Artech House