

Agile Program Management Practices for the Federal Aviation Administration

Version 1.0
October 2016



Approved for Public Release;
Distribution Unlimited. Case Number 16-3811

MITRE

Table of Contents

- Acknowledgments..... 4
- 1 Introduction..... 1-1
 - 1.1 Purpose and Scope 1-1
 - 1.2 Approach..... 1-1
 - 1.3 Document Organization 1-2
- 2 Agile Overview 2-1
 - 2.1 Scrum 2-1
 - 2.2 Scrum Events 2-4
 - 2.3 Scrum Roles 2-4
 - 2.3.1 FAA - Program Management Team..... 2-4
 - 2.3.2 Contractor - Development Team Organization..... 2-6
 - 2.4 Requirements 2-7
 - 2.4.1 Product Vision 2-8
 - 2.4.2 Program Epics 2-8
 - 2.4.3 Features 2-9
 - 2.4.4 User Stories 2-9
- 3 Agile Program Management for the FAA..... 3-1
 - 3.1 Planning 3-1
 - 3.1.1 Program Planning..... 3-1
 - 3.1.2 Release Planning 3-3
 - 3.1.3 Sprint Planning..... 3-4
 - 3.1.4 Planning Challenges..... 3-6
 - 3.2 Development 3-6
 - 3.2.1 Design, Coding, and Integration 3-7
 - 3.2.2 Test and Evaluation..... 3-8
 - 3.2.3 Daily Scrum 3-11
 - 3.2.4 Review and Retrospective..... 3-12
 - 3.2.5 Development Challenges 3-15
 - 3.3 Deployment..... 3-16
 - 3.3.1 Purpose..... 3-16
 - 3.3.2 Deployment Factors and Practices 3-16
 - 3.3.3 Deployment Pipeline..... 3-16
 - 3.3.4 Roles and Responsibilities 3-17

3.3.5	Deployment Challenges	3-17
3.4	Special Topics	3-18
3.4.1	Contract Management	3-18
3.4.2	Estimation	3-20
3.4.3	Agile Metrics	3-21
4	Summary and Additional Considerations	4-1
4.1	Agile PM Summary	4-1
4.2	Agile PM Additional Information.....	4-2
5	References.....	5-1

List of Figures

Figure 2-1. Generic Agile Scrum Development Life Cycle	2-3
Figure 2-2. Agile Program Management Team	2-5
Figure 2-3. Contractor Agile Development Team	2-6
Figure 2-4. Proposed Agile Requirements Hierarchy	2-8
Figure 3-1. Agile Sprint Development Process	3-7
Figure 3-2. Sprint Backlog.....	3-7
Figure 3-3. Test and Evaluation Pipeline.....	3-10
Figure 3-4. Sprint and Release Levels Definition of Done	3-10
Figure 3-5. Sprint Burn-down Chart.....	3-22
Figure 3-6. Sprint Burn-up Chart.....	3-23
Figure 3-7. Defect Burn-up Chart.....	3-24
Figure 3-8. Velocity.....	3-24

List of Tables

Table 1-1. Waterfall vs. Agile Program Management	1-1
Table 2-1. Agile Values and Principles	2-1
Table 2-2. Roles and Responsibilities for Program Management Team.....	2-5
Table 2-3. Roles and Responsibilities for Contractor Development Team.....	2-7
Table 3-1. Program Planning Outputs.....	3-2
Table 3-2. Program Planning Roles and Responsibilities	3-2
Table 3-3. Release Planning Outputs.....	3-3
Table 3-4. Release Planning Roles and Responsibilities	3-4
Table 3-5. Sprint Planning Outputs.....	3-4
Table 3-6. Sprint Planning Roles and Responsibilities.....	3-5
Table 3-7. Design, Coding, and Integration Roles and Responsibilities.....	3-8
Table 3-8. Roles and Responsibilities for T&E	3-11
Table 3-9. Roles and Responsibilities for Daily Scrum.....	3-11
Table 3-10. Roles and Responsibilities for Sprint Review	3-12
Table 3-11. Roles and Responsibilities for Release Review.....	3-13
Table 3-12. Roles and Responsibilities for Sprint and Release Retrospective	3-14
Table 3-13. Roles and Responsibilities for Deployment	3-17
Table 3-14. Roles and Responsibilities for Contract Management.....	3-19
Table 4-1. Summary of Agile Program Management Considerations	4-1

Acknowledgments

This document is the product of a collaboration between the Federal Aviation Administration (FAA)'s Mission-Oriented Investigation and Experimentation (MOIE) program, an independent research and development (R&D) program sponsored by the FAA and conducted by The MITRE Corporation. The MOIE program is focused on R&D for sponsor mission applications, and this project was selected due to its relevance and potential impact on FAA acquisition, program management, and systems engineering.

1 Introduction

Program management entails the organization and execution of resources to achieve a developmental goal within programmatic constraints such as cost and schedule. For the Federal Aviation Administration (FAA), it provides guidance for how to deliver an operational capability on time and within the budget, while balancing scope, quality, and stakeholder concerns [1]. Program management must be tailored to accommodate variances in projects and organizational culture. One way to tailor this process is to make it more Agile. Agile program management promotes efficient delivery of highly valued and validated capabilities through focused iteration of planning, execution, and monitoring steps in the implementation process. Table 1-1 presents a contrasting view of traditional (Waterfall) program management and Agile program management activities.

Table 1-1. Waterfall vs. Agile Program Management

Traditional (Waterfall) Program Management	Agile Program Management
Single integration	Periodic/Continual Integration
“Success” is measured by delivery of all requirements	“Success” is measured by delivering valuable capabilities
Program management provides estimates upfront	Rough estimate upfront, and continuously refined by the team
Tracks progress by stages for the entire program	Tracks progress by value of capability delivered for fixed time-boxed phases of the program
Structured by lines of businesses	Multidisciplinary teams
Issues discovered later in development life cycle	Issues discovered earlier in development
Adds time/money to release to adapt to issues/requirements	Reduces functionality in release (pushed to future development effort) to adapt to issues/requirements

1.1 Purpose and Scope

This document provides guidance for Program Managers (PMs) executing Agile methodologies. It is specifically targeted for key acquisition stakeholders within the FAA’s Information and Technology organization (AIT), including PMs and contractors, charged with non-National Airspace System information technology (IT) investments and engaged with contractors executing an Agile development process. Given the scope variance of the technical solution and supporting development effort, this guide is also applicable at the project level; the terms program and project are used interchangeability throughout this document.

Practices described in this document are expected to be tailored appropriately for each program’s situation. It is assumed that the program and contractor agree to follow an Agile development process, such as Scrum or Kanban. This document is not intended to provide comprehensive guidance on Agile development (including the criteria for employing Agile techniques) or Agile program management. It is expected to be a living document that will be updated with additional Agile program management guidance as appropriate.

1.2 Approach

The MITRE Corporation evaluated Agile program management techniques used by industry and federal government agencies. We expand upon “Federal Aviation Administration Agile Acquisition Principles and Practices” guidance [2], which identifies Agile principles and practices for FAA acquisition

executives regarding the adoption and use of Agile in the FAA environment. Additional Agile guidance is provided regarding the following aspects of program management:

- Strategic and tactical planning
- Development, including review events, and deployment
- Monitoring and control.

Challenges to the adoption and execution of Agile program management are identified for further discussion.

1.3 Document Organization

- Section 1 addresses document purpose, scope, approach, and provides a framework for discussing how the Agile program management technique is different from the Waterfall technique.
- Section 2 provides an Agile overview and details what is included in the Scrum development process.
- Section 3 provides details regarding the Agile process including recommended best practices, challenges, and the role of the PM for the following topics:
 - Planning
 - Development
 - Deployment
 - Contract Management
 - Cost Estimation and Metrics
- Section 4 concludes the guidance with a summary and additional considerations.

2 Agile Overview

Agile is a collection of techniques, values, and principles that are iteratively and collaboratively applied in a cross-functional fashion to provide early delivery of valued products. The Agile values, expressed in the Agile Manifesto [3] and principles [4], are captured in Table 2-1. The bolded items under Agile Development Values represent items of the highest value. A number of software development methodologies (e.g., for example such as Scrum, Extreme Programming (XP), Kanban, and Lean Development), support the Agile values and principles. The use of Agile is particularly well suited for more challenging development efforts, such as those faced by the FAA, where there is some uncertainty regarding technical requirements and operational needs.

For the differences between Agile and more traditional program management and development approaches, refer to Section 1.3.2 of the “Federal Aviation Administration Agile Acquisition Principles and Practices” [2].

Table 2-1. Agile Values and Principles

Agile Development Values	Agile Development Principles
1. Individuals and interactions over processes and tools	1. Customer satisfaction by rapid delivery of useful software
2. Working software over comprehensive documentation	2. Welcome changing requirements, even late in development
3. Customer collaboration over contract negotiation	3. Working software is delivered frequently (weeks rather than months)
4. Responding to change over following a plan	4. Close, daily cooperation between business people and developers
	5. Projects are built around motivated individuals who should be trusted
	6. Face-to-face conversation is the best form of communication (co-location)
	7. Working software is the principle measure of progress
	8. Sustainable development, able to maintain constant pace
	9. Continuous attention to technical excellence and good design
	10. Simplicity – the art of maximizing the amount of work not done – is essential
	11. Self-organizing teams
	12. Regular adaptation to changing circumstance

It is important to remember that “Agile is a journey, not a destination and the best the organization can hope for is that teams become more Agile by embedding its mindset deeper inside themselves and the organization” [5]. A number of processes and practices, such as Scrum, can help get the organization into the Agile mind-set, and this guide will make recommendations on how to apply them.

2.1 Scrum

The Scrum framework supports Agile values and principles. It is the most popular and widely applied framework in Agile software development when government programs apply Agile.

Scrum as defined by Mike Cohn [6]:

Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.

It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month).

The business sets the priorities. Teams self-organize to determine the best way to deliver the highest priority features.

Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance it for another sprint.

Scrum is good for team organization and management, and it is often complemented with XP development practices, such as paired programming and Acceptance Test Driven Development. It must be noted that Scrum may also be scaled to accommodate large programs that are very complex and multiple teams that may not be co-located.

Figure 2-1 shows the Scrum framework for a generic Agile development life cycle at a very high level. The following sections will decompose the elements of Scrum, describing the associated events and roles as recommended by “The Scrum Guide” [7] for implementing Scrum. Later sections of this guide provide recommendations on how to apply these events and roles within the FAA.

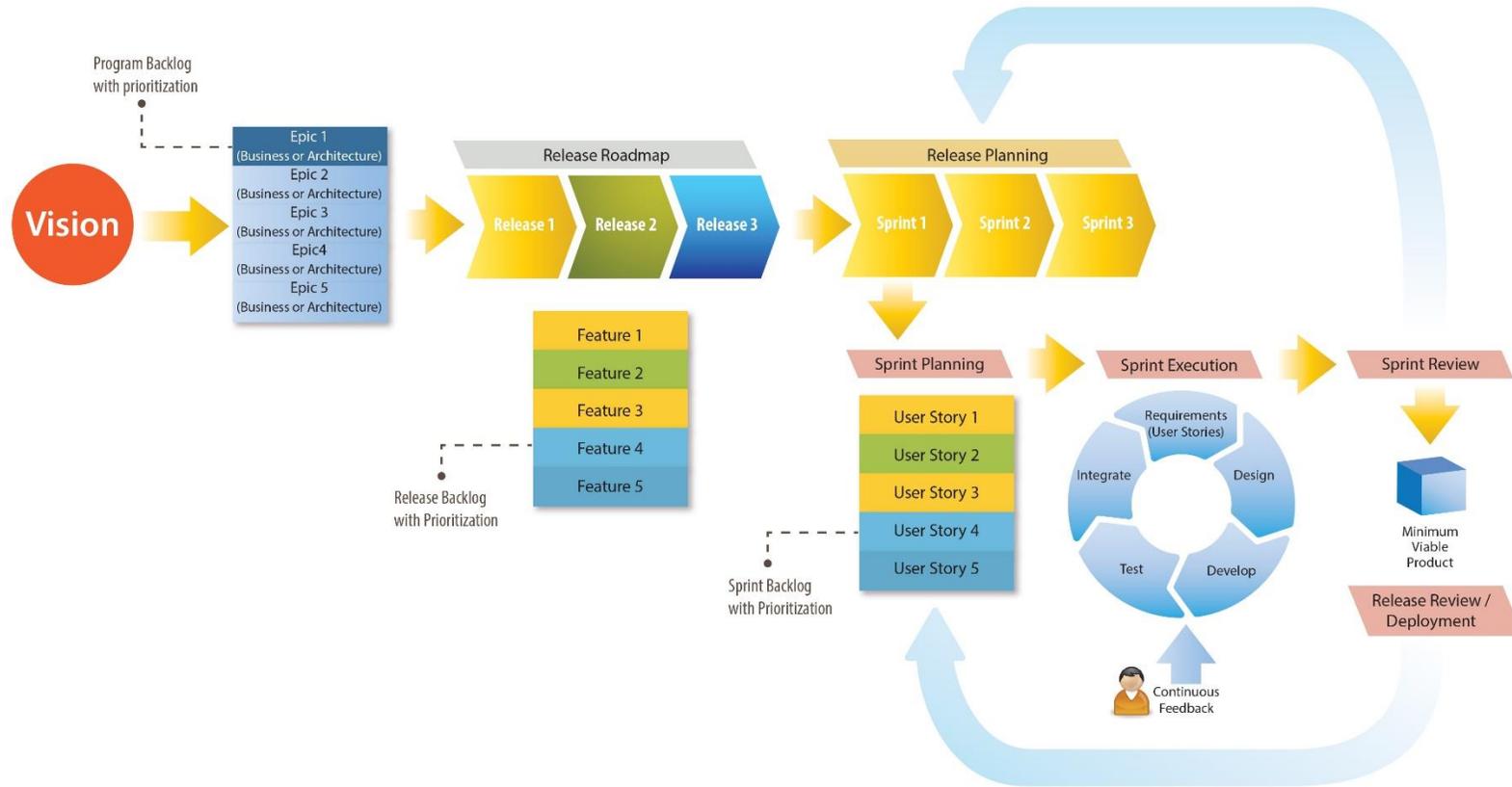


Figure 2-1. Generic Agile Scrum Development Life Cycle

2.2 Scrum Events

Throughout the development life cycle, as represented in Figure 2-1, the following Scrum events will take place:

- **Release:** The Release represents a set of capabilities that may be ready for operational deployment. This package of capabilities contains highly valued features (or the minimum viable product, which is a set of capabilities that the user community has identified as most valuable) that have been designed, developed, integrated, and tested through a series of Sprints.
- **The Sprint:** Potentially deployable capabilities are designed and developed through a series of iterations, or time-boxed periods. The same time-boxed periods are consistently used throughout the development timeframe, and determined based on the Development Team’s capability and the complexity of the operational needs.
- **Planning:** Three levels of planning are promoted within Agile: (a) the program level, which represents a series of projects/development efforts that are conducted concurrently to meet an operational need; (b) the release level; and (c) the Sprint level. Agile planning is strategic at the program level and becomes more tactical at the Sprint level. Planning activities entail laying a specific development plan (or roadmap) for when certain features will be developed and deployed, helping PMs determine what resources are needed at a particular time.
- **Daily Scrum:** The Daily Scrum is a 15-minute standup where the Development Team coordinates and plans the next day’s activities. The intent is to share and measure progress for a given Sprint and to identify any obstacles.
- **Review:** Reviews provide an opportunity for the entire team, including stakeholders (e.g., user community) and developers, to come together and evaluate what was accomplished during the Sprint and Release relative to the Sprint plan and Release Backlog. The developed set of features are demonstrated and feedback is used for subsequent planning events. As the Release leads to deployable capabilities, the Release Review is a prominent event and may have a broader set of stakeholders/participants than a Sprint Review.
- **Retrospective:** Culminating the Sprint, Release, or program is a retrospective. This event is an outcome of the “continuous improvement” mind-set and provides an opportunity to reflect upon the processes, tools, and team organization. Issues and mitigation strategies are identified for implementation in successive Sprints/releases.

2.3 Scrum Roles

The Scrum process has three defined roles that must be part of the FAA’s and contractor’s teams. The following sections describe the program (FAA) and development (contractor) team organization, highlighting the typical roles and responsibilities for individuals and engagement expectations between the government and contractor.

2.3.1 FAA - Program Management Team

The FAA’s Program Management Team needs to be structured to facilitate Agile development. It introduces and defines new roles and responsibilities that are not present during Waterfall development practices. Figure 2-2 highlights the FAA members on the Program Management Team (see blue outline for members) and identifies their roles and responsibilities for an Agile development environment.



Figure 2-2. Agile Program Management Team

Table 2-2 captures the roles and responsibilities for the Program Management Team.

Table 2-2. Roles and Responsibilities for Program Management Team

Role	Responsibility
PM	<ul style="list-style-type: none"> Coordinates with stakeholders (including those within the organization and external, such as the contract or finance organizations), and serves as the key interface to the enterprise and the development team. Builds awareness across communities to help enable agility, and defines clear communication plans for how the FAA and the contractor will engage. Determines the capabilities to be delivered per Release/Sprint, and accountable for on-time delivery performance. Works with senior leaders to determine the root cause when a failure to deliver on time occurs, adapting to changes when necessary.
Lead Engineer	<ul style="list-style-type: none"> Works closely with the PM and Product Owner to oversee the technical aspects of the program. Provides input regarding resource scheduling and ensures that enterprise architecture and/or technical standards are being considered.
Product Owner¹	<ul style="list-style-type: none"> Maximizes the value of the work conducted by the Development Team; serves the Development Team, interfaces with interacting programs, and manages the Sprint Backlog as the authorized representative of the user community. Approves and accepts User Stories (which express requirements in Agile development - see Section 2.4.4 for more information), provide acceptance criteria

¹ In situations where the government Product Owner is not readily accessible, the contractor developer organization may appoint an internal Product Owner, who is a counterpart to the FAA's Product Owner. Coordination necessary to ensure alignment of expectations and direction. Expect that the FAA's Product Owner and PM to have final authority to provide direction to the contractor.

Role	Responsibility
	and the definition of “done,” and provide technical (including architectural input) direction to the team during the Sprints.

2.3.2 Contractor - Development Team Organization

The Development Team members are responsible for the execution of the Agile software development activities required to deliver an increment of functionality at the end of a fixed time period (Sprint). For the FAA, it may be a selected contractor or a Development Team within the FAA. For discussion, this document assumes that the Development Team consists of contractor personnel.

Figure 2-3 specifies the members of the team, highlighting that there are specific roles and responsibilities within the team, based on their technical capabilities. For example, for a Development Team that is comprised of seven plus or minus two members that may have technical expertise not limited to software development, quality assurance, requirements, and system architecture. Development Teams may be scaled to meet the scope of the project, thus, running parallel Scrum development efforts. The Development Team may be supported by additional individuals who have specific technical specializations, such as security, safety, or data analysis. These individuals may comprise a Support Team that works with the Development Team on an as-needed basis and are not involved in the day-to-day development activities. There may be additional roles that are not represented in Figure 2-3. For example, the team may have a PM, who is the FAA’s PM counterpart. (For additional guidance regarding the role of a Scrum Master, see [2].)



Figure 2-3. Contractor Agile Development Team

There is a government-contractor relationship between the FAA’s Program Management Team and the contractor’s Development Team, regardless of the development methodology (e.g., Agile). With Agile, the level of communication and engagement changes through the introduction of the Product Owner, who also works directly with the Development Team. With this change, all FAA personnel should maintain awareness of the policy in the Acquisition Management System on personal services. They should find the balance between providing monitoring/technical direction within the terms of the contract and avoid inadvertently creating an employer-employee relationship with the contractor’s personnel. Table 2-3 captures the roles and responsibilities for the contractor Development Team.

Table 2-3. Roles and Responsibilities for Contractor Development Team

Role	Responsibility
Contractor - PM	<ul style="list-style-type: none"> • Coordinates among FAA PM, FAA Product Owner, FAA Contract Officer and personnel on the Contractor Development Team, serving as a key interface to the program and development team. • Ensures that Development Team completes Release and Sprint goals within the allotted time (e.g., four weeks for a Sprint). Completion includes working software and associated artifacts. • Works with FAA PM and Product Owner for planning and deployment purposes, ensuring alignment in expectations from the Development Team.
Contractor - Scrum Master	As the “servant-leader,” ensures that the Development Team and Product Owner are following Agile practices, and removes/mitigates impediments for the team.
Development Team	<ul style="list-style-type: none"> • Maximizes the value of the work through the delivery of incremental capabilities. • The team (software developers and testers, augmented by Support Team specialists that could include a system architect, configuration manager, requirements engineer, security engineer, and others), executes the Sprint, including participation in planning, development, and deployment activities.

2.4 Requirements

Agile requirements represent the evolving functionality that will address an operational need. Compared to traditional requirements (i.e., shall statements), Agile promotes a different structure to capture requirements. Figure 2-4 represents the requirements hierarchy. Operational needs are captured at a high level through a Product Vision, and then decomposed into Epics, Features, and User Stories. The FAA is accountable for the Epics, and the contractor is accountable for the User Stories. There is a sense of shared responsibility at the Feature level. Accountability includes any requirement decomposition and management activities. See [2] for more information regarding the management of requirements in the Program, Release, and Sprint Backlogs, where a backlog is a repository for upcoming work. The aforementioned backlogs contain Epics, Features, and User Stories, respectively.

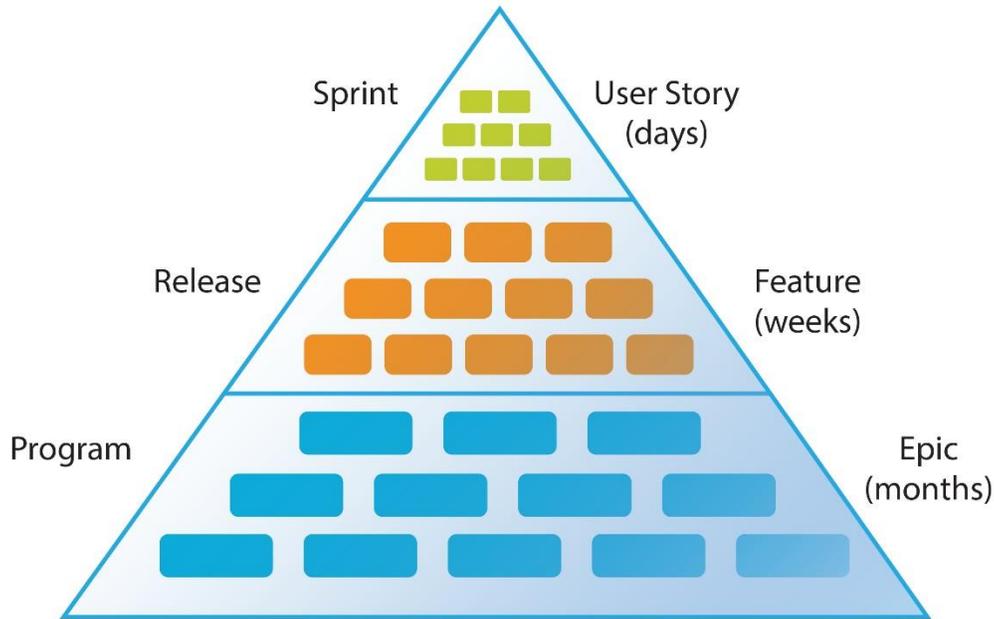


Figure 2-4. Proposed Agile Requirements Hierarchy

2.4.1 Product Vision

The Product Vision is a future view of the solution, and provides the larger, contextual overview and purpose of the solution under development. The Product Vision should:

- Reflect user and stakeholder needs
- Include primary capabilities or features proposed to address the needs; and
- Describe the boundaries and constraints that influence the solution and acquisition decisions to be made.

Sources of the Product Vision include operational shortfalls, agency strategies and business plans, enterprise portfolios, and architecture roadmaps. The FAA, including the PM, is responsible for defining the Product Vision. The typical format for a Product Vision includes the following information:

- For (target customer)
- Who (statement of the need or opportunity)
- The (product name) is a (product category)
- That (key benefit, compelling reason to buy)
- Unlike (primary competitive alternative)
- Our product (statement of primary differentiation) [8].

2.4.2 Program Epics

The program Epics communicate the business needs and goals for specific products generated by the program. They are generated from the Product Vision, and used to formulate the cost basis for the program; describe functionality, constraints (non-functional requirements), and desired architecture; and define the minimum viable product, which is a representation of highly valued capabilities that form the technical baseline. Program Epics are managed in the Program Backlog, arranged into realizable, prioritized groupings, which can be managed to achieve timely functionality. Management at this level consists of flexibility of priority (scope) and schedule. At this stage, the FAA PM has responsibility of

creating the program Epics and maintaining the Program Backlog, working with the Business Owners to elicit program Epics and with the Contracting Officer to define the solicitation for potential bidders based on information captured in the program Epics. The scope flexibility at this level is expected to be limited. However, deletion may be necessary due to events, and substitutions may be made as supported by cost-free clauses. Program Epics may be equivalent to the System Specification Document (SSD), as it captures the requirements in a single location.

2.4.3 Features

The Features capture program-level functionality provided by a system to fulfill stakeholder needs. Features are derived from the program Epics (a one-to-many traceability) and used to define the minimum viable product in greater detail for planning purposes. Features are generated to build an understanding and consensus between the FAA and the selected contractor regarding the functional design, meaning that both parties have a role in the generation and management of Features and that there is no loss in fidelity during the transition from the FAA to the contractor. The Release Backlog contains the list of Features, and is prioritized to reflect the Features that have been selected for development as part of the Release. Features, along with program Epics, may be equivalent to the SSD requirements document that the FAA typically produces prior to the commencement of development work.

2.4.4 User Stories

A User Story represents desired functionality, expressed in a concise, simple description from a user's perspective. User stories are typically developed by the Development Team (contractor), but the FAA PM may wish to evaluate them from a programmatic perspective (evaluate status and resource needs). They may be equivalent to the requirements captured in the System/Subsystem Specification and/or Software Requirements Specification documents, which detail the software design captured by the contractor during development. The Sprint Backlog can be used to manage the User Stories. User Stories are derived from Features, test capabilities, and identified constraints. They have the following characteristics:

- Are expressed in the business language of the user
- Capture who, what, and why
- Represent a small piece of business value that be realized in a development iteration
- Contain just enough information so that the developers can produce a reasonable estimate of the effort to implement them
- Include criteria to determine when the User Story development is complete (definition of done and acceptance criteria); and
- Can be prioritized and collected in a Sprint Backlog.

More specifically, User Stories should at a minimum contain the following elements:

- **The basics:**
 - Name
 - Description: As a [user] I want [action] so that [benefit]
 - Completion/acceptance criteria
 - Priority/rank
 - Effort estimate (For example, story points may be used to describe the relative complexity or the relative effort of work to implement a User Story. See Section 3.4.2 for information.)
- **Other information:**
 - Unique identifier (potentially structured to relate to pertinent Epic and Feature)

- Source and date of creation
- Status
- Change history
- Notes and supporting information

3 Agile Program Management for the FAA

This section addresses the following aspects of Agile program management:

- Planning
- Development
- Deployment
- Special Topics, such as Contract Management and Estimation.

3.1 Planning

An Agile program conducts planning at three levels:

- Program planning is strategic in nature and defines the overall context of the program, including key capabilities that will be produced by the program in the Program Backlog. This level of planning entails defining a product vision and developing a roadmap consistent with enterprise-level objectives for the program.
- Release planning is both strategic and tactical in nature. This level of planning focuses on how a program can incrementally deliver value through major releases over time. Release planning occurs at a regular cadence, which is commensurate with the duration of Release development.
- Sprint planning is tactical in nature. This level of planning focuses on identifying the list of Features that will be developed during a particular Sprint.

Estimation is a key part of planning and occurs at all levels, becoming more refined at the sprint level. Further discussion may be found in Section 3.4.2.

3.1.1 Program Planning

Program planning is a strategic-level activity that focuses on setting the overall context of the program and defining the major capabilities to be delivered consistent with enterprise objectives. The key inputs for program planning are:

- Customer/user needs
- Input from key stakeholders
- Agency strategy and business plans
- Enterprise-level documentation (e.g., portfolios, enterprise architecture roadmaps).

A key output of the program planning process is the product vision, which articulates the context and purpose of the solution under development. Epics, which comprise the high-level functionality delivered by the program, align with the product vision, and are organized into a Program Backlog. The product roadmap helps to communicate the implementation plan and schedule for the program. Even the initial stages of program planning may entail the development of Features, through interviews with users and key stakeholders.

Table 3-1 presents further details on the outputs for program planning.

Table 3-1. Program Planning Outputs

Output	Description
Product Vision	A product vision defines the operational context and purpose for the solution under development. It describes the future view of the solution to be developed to provide the larger, contextual overview and purpose of the solution. See Section 2.4.1 for more information.
Program Epics	Program Epics communicate the enterprise needs and goals for the product generated by the program. They describe the functionality, and potentially, the desired architecture. See Section 2.4.2 for more information.
Program Backlog	The Program Backlog is an ordered list of Epics allocated to a program, which can be reorganized to reflect changes in priorities. Given that this may form the technical baseline, it is not expected that program Epics in the Program backlog would be changed. If an Epic is removed, it may be feasible to replace the original Epic with a new Epic by using a cost-free contractual clause.
Product Roadmap	The product roadmap describes the planned evolution of the product based on strategic priority. Specifically, it identifies the Epics to be developed during the implementation timeframe. The roadmap provides a sense of direction and lays out a realistic plan to fulfill the product vision. It helps with communication of purpose, alignment of expectations, and prioritization; however, it should not be considered a fixed plan or commitment.
Features	Features describe the functionality provided by a product to fulfill stakeholder needs. See Section 2.4.3 for more information.
Test and Evaluation Strategy	The strategy will be reflected in products such as the Implementation Strategy Planning Document (ISPD) and the Test and Evaluation Master Plan (TEMP), which documents the overall structure and objectives of the test and evaluation program. The Statement of Work (SOW) will impose requirements on the developer to plan and conduct a test program, including documentation such as the contractor TEMP and test plans, procedures, and reports.

Until a program has been instantiated or “stood up,” the Business Owner is responsible for program planning. Once the program has been instantiated, the PM takes over. In general, the PM and Product Owner play significant roles during the program planning process. Table 3-2 defines specific roles and responsibilities for program planning.

Table 3-2. Program Planning Roles and Responsibilities

Role	Responsibility
FAA - Business Owner	Develops the product vision for the intended solution. Product visions originating at an enterprise level should be developed by the relevant enterprise organization. In addition, the Business Owner provides input into the program Epics, User Stories, and product roadmap, serving as the final acceptance authority for the solution delivery.
FAA - Product Owner	Provides input into the product vision and the product roadmap. Also collaborates with the PM to develop and manage the program Epics and

Role	Responsibility
	Program Backlog, ensuring that the operational needs (as defined by the users and stakeholders) are captured and derived at the Feature and User Story level.
FAA - PM	Oversees and manages program planning. The PM facilitates the development of each of the products associated with program planning as necessary, including the product vision, program Epics and the Program Backlog. The PM develops and manages the product roadmap.
User and/or Stakeholder Community	Collaborates with Product Owner, and provides input into the product vision, program Epics, and product roadmap.

3.1.2 Release Planning

Release planning is both strategic and tactical in nature, and focuses on the incremental delivery of releases for the solution under development under a regular cadence. As a Release encompasses multiple Sprints, Release planning should focus on the development activities for about the next three to six months (note that Office of Management and Budget recommends six-month delivery increments [9]). PMs must tailor the duration of a Release to the development cadence, ensuring that the Features are properly decomposed from Epics, and that the team is able to develop potentially deployable functionality by the end of the Release. The key inputs for release planning are:

- Product Vision
- Program Backlog
- Product Roadmap.

The major outputs of the Release planning process are the Features and the Release Backlog. Features identify, at a lower level of detail than program Epics, specific functionality needed to meet user and stakeholder needs. These Features are organized into a Release Backlog, which specifies the Features to be developed for a particular Release. Finally, the Program Backlog is also revised as necessary as part of the Release planning process.

Table 3-3 provides further details on the outputs for release planning.

Table 3-3. Release Planning Outputs

Output	Description
Features	Features describe the functionality provided by a product to fulfill stakeholder needs. See Section 2.4.3 for more information.
Release Backlog	The Release Backlog comprises a set of Features that have been selected for development as part of a Release. It is documented in priority order, and can be revised based on changing priorities. The Release Backlog identifies the minimum viable product to be deployed to users at the end of a time-boxed period, based on the program’s Agile cadence.

In general, the PM and Product Owner play significant roles during the Release planning process. Table 3-4 defines specific roles and responsibilities for Release planning.

Table 3-4. Release Planning Roles and Responsibilities

Role	Responsibility
FAA - Product Owner	Collaborates with the PM and Development Team to decompose the Epics, synthesize User Stories, and define the Features associated the Epics. The Product Owner collaborates with the PM and Development Team to develop and manage the Release Backlog.
FAA - PM	Oversees and manages Release planning. The PM facilitates the development of each of the products associated with Release planning as necessary, including the Features and Release Backlog. May need to collaborate with the Contractor PM, if identified, during Release planning.
Business Owner, Users, and Stakeholders	Provide input into Feature definition and the Release Backlog as necessary.
Contractor - Development Team	Provides input into Feature definition and the Release Backlog as necessary, including providing input regarding estimation and resource availability from a Release perspective.

3.1.3 Sprint Planning

Sprint planning is tactical in nature. This level of planning focuses on identifying the list of Features that will be developed during a particular Sprint and understanding all of the work necessary to develop those Features in a fixed period of time (e.g., two to four weeks). PMs must tailor the duration of a Sprint to the development cadence, ensuring that the User Stories, including the associated tasks, are workable by the Development Team within the iteration. The key inputs for Sprint planning are:

- Product Vision
- Program Backlog
- Release Backlog.

The major outputs of the Sprint planning process are the User Stories and Sprint Backlog. The Sprint Backlog comprises the User Stories and tasks to be completed during the Sprint in priority order. User stories are refined to understand the functionality that needs to be implemented during the Sprint. The Release Backlog may be revised as necessary. Note that the architecture infrastructure may be addressed in a “Sprint Zero”, where the underlying infrastructure may be initially defined prior to beginning any software development that will reside on top of the architecture.

Table 3-5 defines further details on the outputs for Sprint planning.

Table 3-5. Sprint Planning Outputs

Output	Description
User Stories	The User Stories represent a small piece of business value to be realized in a Sprint. User stories are refined to identify discrete elements of functionality related to a defined Feature with explicit completion/acceptance criteria that can be implemented within the Sprint. They contain just enough information so that the developers can produce a reasonable estimate of the effort to implement them, and include criteria to determine when the User Story development is finished. Actual estimates from previous Sprints may inform the

Output	Description
	plan for the next Sprint, with regard to the team’s ability. See Section 2.4.4 for more information.
Sprint Backlog	The Sprint Backlog comprises a set of User Stories derived from the Features and Release Backlog that have been selected for development as part of a Sprint. It documents User Stories in priority order, and consists of the set of functionality to be developed during the Sprint. The Sprint Backlog identifies all the work the Development Team identifies as necessary to develop the necessary functionality. It is revised throughout the Sprint to reflect the work required to complete the Sprint, as the Development Team learns more. The Sprint Backlog represents a real-time view of the work the Development Team plans to accomplish during the Sprint.

In general, the PM, Product Owner, and Scrum Master play significant roles during the Sprint planning process. Table 3-6 defines specific roles and responsibilities for Sprint planning.

Table 3-6. Sprint Planning Roles and Responsibilities

Role	Responsibility
FAA - Product Owner	Collaborates with the PM and Development Team to identify the Release Backlog items selected for the Sprint, and identify priorities for the Sprint Backlog. They provide input on the “definition of done” (User Story attributes that define the quality standards to ensure completion of a User Story. An example User Story for a movie provider service will state: <i>As a customer, I can search for movies by title so that I view what movies are available for my personal viewing.</i> The associated Acceptance Criteria specifies that the functionality described by the User Story must 1) be easy to read and understand, and 2) have fast response times (<5 seconds). The associated Definition of Done describe how to ensure quality of the functionality, and specifies that the Development Team must have 1) perform and pass unit tests, 2) commit the source code, and 3) complete functional review with user community/Product Owner) for the Sprint as the acceptance authority for User Stories. The Product Owner also updates the Sprint Backlog and User Stories as necessary.
FAA - PM	Oversees Sprint planning.
Contractor – PM	Works with the FAA PM to establish the goals for each Sprint, and to ensure that the resources are available to complete the agreed-upon Sprint schedule.
Contractor - Scrum Master	Facilitates Sprint planning.
Contractor - Development Team	Executes Sprint planning. The Development Team develops the User Stories based on the Features and the Release Backlog. It also develops and manages the Sprint Backlog.
Business Owner, Users and Stakeholders	Do not participate in Sprint planning directly

3.1.4 Planning Challenges

The following questions illustrate the challenges associated with planning, and should be evaluated prior to and during the planning process.

1. Has the program identified a Product Owner?
 - The Product Owner should be identified within the FAA’s organization to have the decision authority and power to provide technical direction to the contractor.
 - Subject Matter Experts may be brought in as a technical advisor, but may not have the authority to represent the FAA (no decision authority or ability to provide direction).
2. Has the FAA PM and PO received sufficient training or expertise with Agile methodologies to facilitate Agile planning adequately? What are the key gaps, if any? What steps need to be taken to address the gaps?
 - The Agile planning process requires a significant amount of knowledge about Agile methods at multiple levels, and should not be undertaken without sufficient expertise. Each of the key individuals involved in the planning process should be trained in Agile methods.
 - Novices to the Agile approach should consult coaches and veterans of successful Agile programs for guidance. In addition, where feasible, exemplar Agile artifacts should be used and consulted to level-set expectations for outputs.
3. How will a “minimum viable product” be defined?
 - The minimum viable product constitutes the minimum set of Features that comprise an acceptable product. It is important to leverage input from Business Owners, users, and stakeholders while defining the minimum viable product.
4. Agile planning (and Agile approaches in general) require significant collaboration between the program, the Business Owner, and the Development Team. What challenges are associated with this? What should be considered to address these challenges?
 - The ability to collaborate regularly and face to face to the degree feasible is an important aspect of the Agile approach. This high degree of daily cooperation may not be possible given the current organizational structure and disperse location of personnel, and it is important to take measures to facilitate collaboration.

3.2 Development

From the planning sessions, the Development Team begins to take over the design, development, integration, and test and evaluation (T&E) activities, as shown in Figure 3-1. The following sections detail the related activities, from design activities that may occur during planning to the retrospective. The goal of every Sprint is to produce a potentially deployable product.

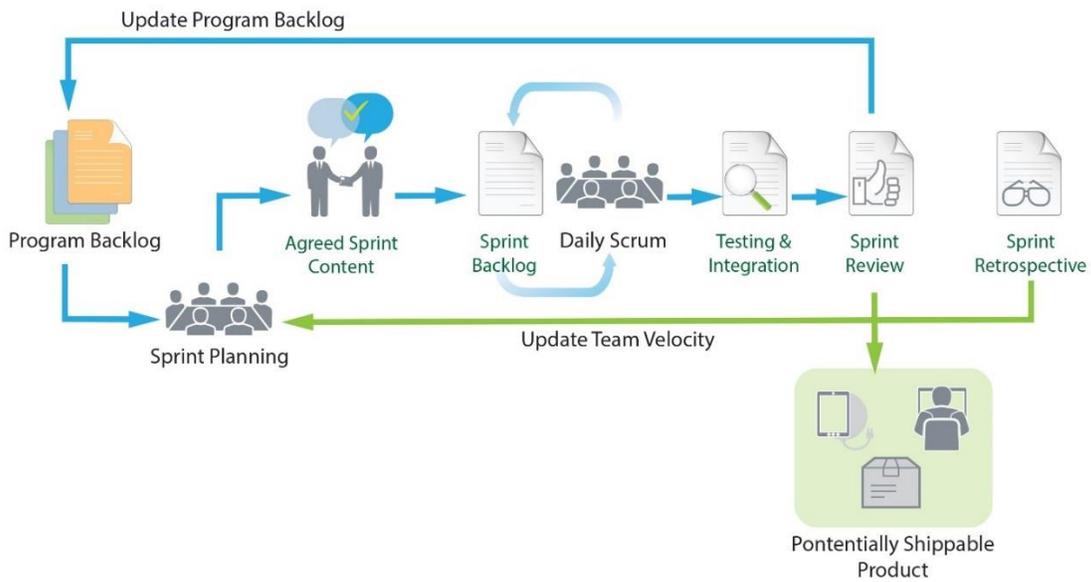


Figure 3-1. Agile Sprint Development Process

The Sprint Backlog, provides a more detailed subset of the Program Backlog. The Sprint Backlog provides every team member with a graphical representation of the status of the tasks being worked in the current Sprint by identifying the tasks required to complete the required User Story. It may be managed through a Kanban board (Figure 3-2), which shows what still needs to be done, what is in progress, and what is completed. The Scrum Master from the contractor Development Team is responsible for maintaining the backlog, and the Product Owner will provide feedback on the User Stories/tasks, including any prioritization and clarity on the acceptance criteria. The Product Owner or FAA PM may also review the Sprint Backlog to gain an understanding of the Development Team’s efforts to determine progress.

SPRINT BACKLOG				
Committed Backlog Items	Tasks not Started	Tasks in Process	Tasks Tested	Completed Tasks

Figure 3-2. Sprint Backlog

3.2.1 Design, Coding, and Integration

In a traditional Waterfall development, the system is completely designed before software development begins (i.e., “big design up front”). In Agile development the design is emergent; it evolves with each Sprint within bounds defined by the roadmap and system architecture. Agile involves a hierarchical approach in which the design is progressively defined from the overall architecture, to the Release, and

finally to the current Sprint. The Development Team has greater autonomy in executing these tasks in the Sprint backlog, working on developing the User Stories around the Sprint cadence. As shown in Figure 3-1, integration occurs earlier and more episodically compared to traditional development cadences (i.e., Waterfall). The Sprint cadence requires integration of developed capabilities for demonstration at Sprint completion. Table 3-7 contains the roles and responsibilities for design, coding, and integration.

Table 3-7. Design, Coding, and Integration Roles and Responsibilities

Role	Responsibility
Design	
Contractor -Development Team	Responsible for the technical solution, including all aspects of the system/software design.
Contractor -Architect	Ensures that the emergent design realizes and is consistent with the planned system architecture.
FAA -Architect/Lead Engineer	Ensures that the contractor design meets overall program roadmap and architecture objectives.
Contractor - Scrum Master	Ensures that design choices are effectively coordinated within and among Development Teams.
Coding	
Contractor -Development Team	Responsible in either Waterfall or Agile for performing assigned development tasks and complying with defined procedures/ practices.
Contractor - Scrum Master	Ensures that impediments to development are overcome.
FAA - Product Owner	Supports developers as necessary to define “done” and acceptance criteria for features under development.
Integration	
Contractor -Development Team	Responsible for integrating developed code during the Sprint.
Contractor - Scrum Master	Ensures that impediments to integration are overcome.

3.2.2 Test and Evaluation

T&E is the process by which a system or components are compared against requirements and specifications through testing. The results are evaluated to assess the progress of design, performance, supportability, etc. Developmental test and evaluation is used to assess specific compliance with requirements. Operational test and evaluation (OT&E) is the actual or simulated employment by typical users of a system under realistic operational conditions.

Agile principles influence the Agile approach to T&E in several ways. The first principle is reflected in the importance of up-front engagement and collaboration of the test community with the stakeholders and Development Team during strategic planning. Ideally, testers will be integrated with the Development Team in the earliest stages of planning and continue to be integral throughout development and implementation. For larger efforts, it may be necessary to have distinct test teams working in parallel with the product Development Team. Test personnel need to be involved early in the planning stages to ensure that testing perspectives are captured in the estimation of required resources, in scheduling, and in risk identification and mitigation. Testing perspectives will be captured in planning documents such as the ISPD and the TEMP, and the developer’s expectations will be reflected in the SOW. At a tactical level,

testers work with users (including the Product Owner, who may provide input and coordinate user engagement) to ensure that User Stories have a definition of “done” and acceptance criteria that are testable. Testers may define test User Stories to encompass the work needed to develop tests and procedures and work with the Product Owner to ensure that priorities reflect the dependencies of testing on software completion. Testers are actively involved with users in ensuring the operational suitability of a delivered Release prior to deployment.

T&E is an integral part of solution implementation and involves evaluating a product from the component level, to stand-alone system, integrated system, and if appropriate, system-of-systems and enterprise levels. In the Agile model, testing occurs in line with development during Sprints, which define the pace of software development and testing. Several levels of testing are involved. Initial testing occurs at the software unit or component level, the lowest level of testing, as the code is developed. Unit tests are usually written by developers as they work on code to ensure that the specific function is working as expected. The definition of “done” includes passing the associated test, enabling credit to be claimed for the software function embodied in the unit. These unit tests may occur at any point in the Sprint as the software unit developments progress. A best practice is to test the code each time it is checked in (i.e., continuous test and integration). Depending on the organization’s expectations for software development, unit testing might include static code analysis, data flow analysis, metrics analysis, code coverage analysis, or other software verification practices. Unit integration and associated integration tests occur episodically, with some organizations conducting integration on a daily basis (e.g., nightly builds). Agile discipline often requires that completed units for which credit is to be claimed be integrated into the software baseline at or before completion of each Sprint. This will involve a higher order of testing, ideally automated, to ensure that new or modified functions/features integrate effectively with existing system functionality. The Agile process culminates in a system demonstration, which provides opportunities for user acceptance and feedback regarding system effectiveness and suitability. The early user exposure and feedback differentiates Agile from traditional T&E—it effectively combines elements of traditional T&E and OT&E, sometimes characterized as combined test and evaluation. The process is illustrated in Figure 3-3.

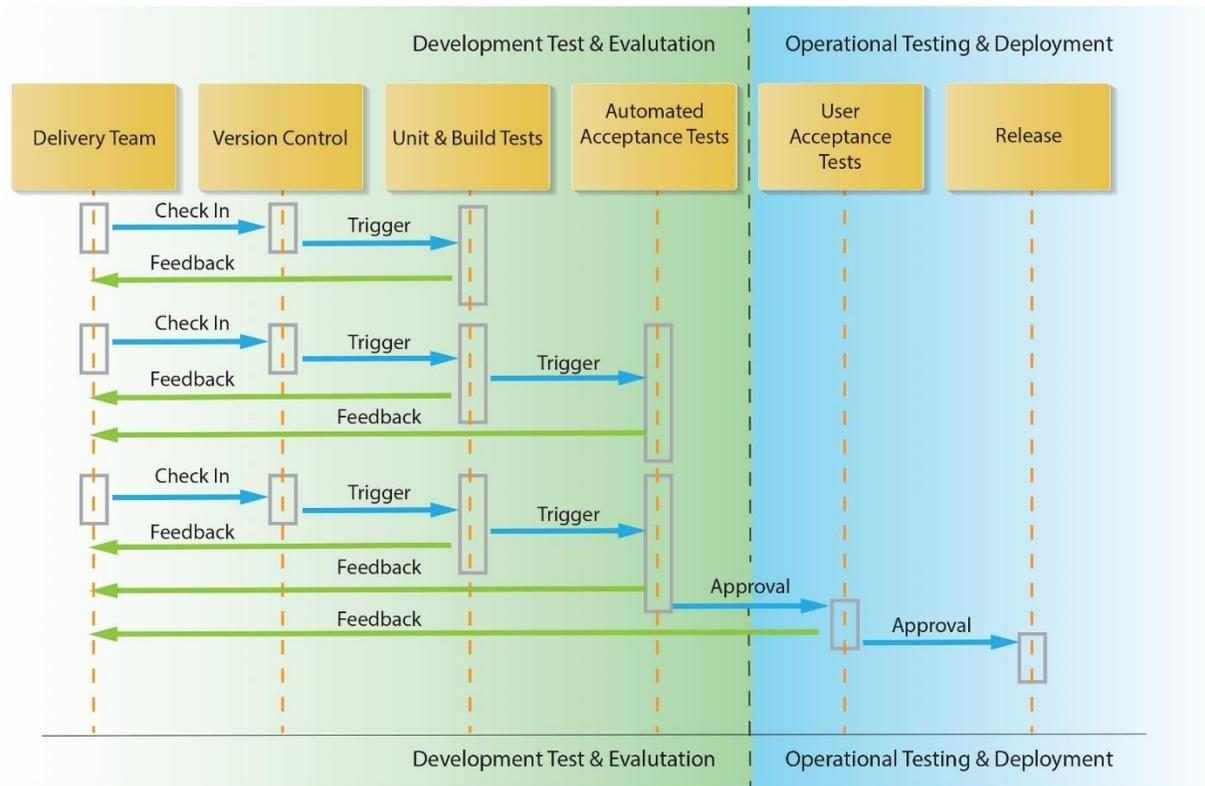


Figure 3-3. Test and Evaluation Pipeline

A critical aspect of this process and the Agile approach in general is the focus on the definition of done (i.e., acceptance criteria). Figure 3-4 depicts factors involved in the definition of done at both the Sprint and Release levels.

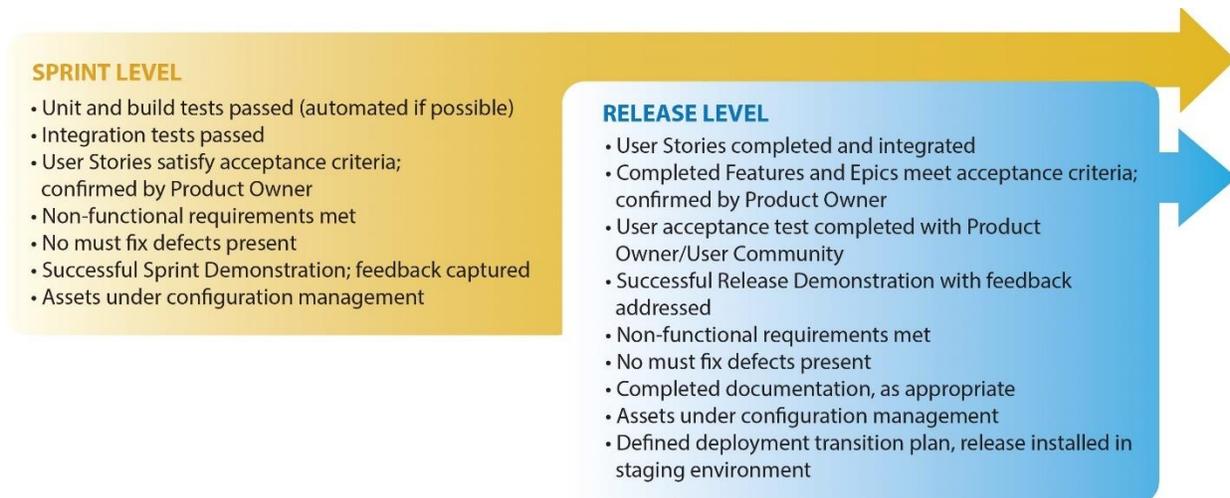


Figure 3-4. Sprint and Release Levels Definition of Done

Table 3-8 presents the roles and responsibilities for T&E.

Table 3-8. Roles and Responsibilities for T&E

Role	Responsibility
Contractor – Development Team	The Development Team, which includes testers, defines test procedures and work with the Product Owner to define acceptance criteria.
Contractor - Scrum Master	Ensures that impediments to testing are overcome.
FAA - Lead Engineer/Test Lead	Formulates T&E strategy consistent with ISPD, documents the strategy in the TEMP, and plans and conducts user acceptance tests.
FAA - Product Owner	Coordinates with Development Team to define test acceptance criteria, determine priorities for resolution of problems identified in testing, and coordinate user availability for user acceptance tests.
FAA – Program Manager	Allocates sufficient resources for T&E.

3.2.3 Daily Scrum

During development, the contractor Development Team will commence writing software and reporting on its progress by a series of Daily Scrum meetings. (Multiple development teams are independent scrum efforts running in parallel, where there is a scrum of scrums daily meeting for coordination and alignment among the teams.) These meetings, in-person or via teleconference, are aimed at coordinating and planning the Development Team’s activities for the coming day. The purpose of the meeting is to improve team communications and ensure that everyone on the team is aware of the team’s progress toward the goals they agree upon during the Sprint planning meeting. The meeting should be short, about 15 to 20 minutes, each day of the Sprint, and should address these three questions:

1. What did I do yesterday that helped the Development Team meet the Sprint goal?
2. What will I do today to help the Development Team meet the Sprint goal?
3. Do I see any impediment that prevents me or the Development Team from meeting the Sprint goal?

The intention of the Daily Scrum is not to solve problems in real-time, but rather to identify issues that need resolution, or whether the Scrum Master needs to mitigate impediments for the Development Team. Table 3-9 captures the Daily Scrum’s key roles and responsibilities for FAA and contractor personnel.

Table 3-9. Roles and Responsibilities for Daily Scrum

Role	Responsibility
Contractor – Scrum Master	Facilitates the Daily Scrum and ensures focused discussion through a short time-boxed period.
Contractor – Development Team	<ul style="list-style-type: none"> • Participates in the Daily Scrum and provides status updates. • Provides input on who should be invited to the Daily Scrum if necessary.
FAA - Product Owner	Participates in Daily Scrum.
FAA – PM	Has no defined role, but may provide input if invited to the Daily Scrum.

3.2.4 Review and Retrospective

The review and retrospective signify the culmination of a Sprint or Release. These Agile events provide the Agile Team, including the contractors and FAA personnel, a means to demonstrate and assess working capabilities. The following sections describe the process and roles and responsibilities for the review and retrospective.

3.2.4.1 Sprint Review

The Sprint Review demonstrates the results of a time-boxed development effort, soliciting feedback from users. The Sprint Review focuses on inspecting a set of User Stories that were developed during a Sprint. It is an informal forum for the team, including the contractor’s Development Team, the user community, and FAA’s Product Owner and PM to evaluate the output of a particular Sprint. At the review, completed and working developmental code is evaluated against the goals of the Sprint, which were initially identified during Sprint planning. The participants of the Sprint Review will evaluate the User Stories that were assigned for that completed Sprint and assess the behavior of the working code through a demonstration, providing feedback throughout. Any changes identified during these discussions are captured in the backlog and may be used during subsequent Sprint planning activities. For a one-month (or four-week) Sprint, the Sprint review should be limited to a four-hour, in-person meeting. While physical meetings may be difficult to arrange, virtual alternatives may be sufficient and allow all participants to provide feedback to the demonstrated capability. The Sprint Review is considered a success with the completion of all identified Sprint goals. [10]

Table 3-10 captures the Sprint Review’s key roles and responsibilities for FAA and contractor personnel.

Table 3-10. Roles and Responsibilities for Sprint Review

Role	Responsibility
Contractor - Development Team	Conducts the Sprint Review.
Contractor – PM	<ul style="list-style-type: none"> • Co-leads the Sprint Review with the FAA’s Product Owner (may include establishing the agenda for the review). • Ensures completion of all activities (e.g., development of code) prior to review, working with the FAA’s PM to schedule this Agile event.
FAA - Product Owner	<ul style="list-style-type: none"> • Co-leads the Sprint Review. • Captures feedback that will be placed into the Sprint Backlog. • Collects feedback from the review for planning purposes (may include identifying changes to the backlog).
FAA – PM	<ul style="list-style-type: none"> • Confirms completion for the Sprint Review and declares success of a Sprint Review. • Ensures that the review is successfully completed and provides program status to key stakeholders/decision makers. • Evaluates the Agile metrics to determine the status of the program (e.g., what was not accomplished during the Sprint, how well did the team perform, and what is left in the Sprint Backlog).
User and/or Stakeholder Community	Validates the set of completed Features and provides feedback.

3.2.4.2 Release Review

The Release represents a set of Features that have been developed, integrated, and tested through a series of Sprints. To prepare for deployment to the field, programs typically have some specific review and/or test milestone to ensure that the developed capability is fit for purpose. The Release Review may serve as an input to this traditional milestone. During this event, the user community and key stakeholders have a formal opportunity to inspect the functionality and provide feedback.

Prior to conducting the Release Review, the team must have complete and working code and documentation (including artifacts related to training and logistics), as they will be demonstrated and inspected during the review. The Release Review is a time-boxed event of one to two hours. The Review provides a means for the Development Team to demonstrate the full set of developed and integrated capability in the context of its operational environment, and discuss solution readiness with the FAA. It is important to stress that a successful Release is more than just a demonstration of the code and artifacts. Having a discussion with all parties, both FAA and contractor, about these Release Review inputs is critical.

Inspection and adaptation are fundamental practices for Agile, and discussions during the Release Review provide the necessary feedback for Release and Sprint planning, including any system behavior modifications and/or changes in priorities. Once the Product Owner and PM review and assess that commitments made in the release plan (or objectives) are satisfactory from a usability and contractual perspective, then the Release Review is complete and the Release is ready for deployment. Upon the completion of the Release Review, unapproved Features that are part of the minimum viable product must continue to be iterated upon by the Development Team until the release is acceptable for deployment. Should the failed Feature not be high priority (or part of the minimum viable product), the FAA may deploy the release without the failed feature. The FAA PM must evaluate the consequences of failure to reach the next stage, Deployment, should the team not successfully pass the Release Review. The Release Review and retrospectives may help the FAA and contractor teams determine appropriate action and steps needed to mitigate the risk of delay. [11], [12]

Table 3-11 captures the Release Review’s key roles and responsibilities for FAA and contractor personnel.

Table 3-11. Roles and Responsibilities for Release Review

Role	Responsibility
Contractor - Development Team	Conducts the Release Review.
Contractor – PM	<ul style="list-style-type: none"> • Co-leads the Release Review with the FAA’s Product Owner (may include establishing the agenda for the review). • Ensures completion of all development activities (e.g., development of code and artifacts during prior sprints) prior to the review.
FAA - Product Owner	<ul style="list-style-type: none"> • Co-leads the review along with the Contractor PM. • Collects feedback from the review for planning purposes (may include identifying changes to the backlog).

Role	Responsibility
FAA – PM	<ul style="list-style-type: none"> • Confirms completion for the Release Review and declares success of a Release Review. • Ensures that the review is completed and provides program status to key stakeholders/decision makers. • Ensures that the Release is operationally coordinated and validated by all necessary users and stakeholder groups and ready for deployment.
User and/or Stakeholder Community	<p>Validates the set of completed Features and provides feedback.</p> <p><i>Note that this group includes unions, any executive sponsors, or representatives from interfacing FAA programs.</i></p>

3.2.4.3 Retrospective

The retrospective is the culmination of a Sprint and Release, taking place after the Sprint and Release Reviews. The retrospective is based on the continuous improvement mind-set, where the team seeks to identify opportunities to improve performance during the next Sprint/Release effort. This is a self and team reflection on the people, relationships, processes, and tools. It is typically fixed to a three-hour meeting for a one-month Sprint. While in-person meetings are preferred when discussing a self/team evaluation, it is possible to leverage videoconferencing capabilities to ensure that all team members can participate.

Participation in the retrospective is dependent on the team’s preference; however, it is a best practice to include each of the following: the program (members of the FAA’s Program Management Team, including the PM, Product Owner, Business Owner, Contracting Officer (CO) / Contracting Officer’s Representative (COR), and integration and test personnel), the Development Team (the contractor team at a minimum, and possibly the Product Owner), and a joint FAA and contractor representative.

As the Development Team is primarily charged with conducting the Sprints, it will be focused on conducting the Sprint retrospectives. As the FAA focuses on the programmatic and deployment issues, it will be focused on conducting the Release retrospectives. A joint retrospective addresses any issues that arise during contractor and FAA interactions/communications. Regardless, discussions during these meetings should focus on what went well and on identifying issues and mitigation plans (or suggested changes for improvements).

Table 3-12 captures the Sprint and Release retrospective’s key roles and responsibilities for FAA and contractor personnel.

Table 3-12. Roles and Responsibilities for Sprint and Release Retrospective

Role	Responsibility
Contractor – PM	<ul style="list-style-type: none"> • Coordinates and conducts the Sprint retrospective (must confirm completion of event). • Coordinates with FAA PM for the joint retrospective. • Ensures that the team is committed to executing the actionable improvement plans during the next Sprint/Release.

Role	Responsibility
Contractor Development Team	Participates in Sprint retrospective and provide feedback on who should be invited to the Retrospective and feedback for improvement.
FAA – PM	<ul style="list-style-type: none"> • Coordinates and conducts the Release retrospective (must confirm completion of event). • Coordinates with contractor PM for the joint retrospective. • Ensures that the team is committed to executing the actionable improvement plans during the next Sprint/Release.
FAA - Product Owner	Participates in Sprint retrospective and provide feedback on who should be invited to the Retrospective and feedback for improvement.
FAA – Program Management Team	Participates in Sprint retrospective and provide feedback on who should be invited to the Retrospective and feedback for improvement.

3.2.5 Development Challenges

There are some challenges associated with development. The PM must evaluate and answer the following prior to execution:

1. How is the “definition of done” defined for a Release? What needs to be satisfied (completed) in order for working software to be deployed?
 - With regard to release and Sprint reviews, PMs and COs must identify what artifacts, including any documents that are in a typical Contract Data Requirements List, are of highest value and should be required as input to the Release Review.
 - Example artifacts may include training manuals, an architecture diagram, and requirements (e.g., system specification or interface) documents. The required documentation may reflect the completed set of User Stories for a particular Release and serve as the Release technical baseline. Determining the value of artifacts, and their intended use, may be inputs into identifying those that are required as part of the Release Review.
2. What is the right level of design documentation to meet program needs?
3. How does a part-time Product Owner meet their responsibility during development (including integration and T&E)?
4. How do the contractor and the FAA ensure design coherence and integrity as the design evolves?
5. How do the FAA and the contractor manage the complexities of integration testing and system-of-systems testing?
 - Integrating functions during Sprints may involve only project-specific assets, with external interfaces driven by simulations. Exposure to operational interfaces may not occur until testing a release in preparation for deployment or, in the extreme case, until the release becomes initially operational.
 - What about new/updated automated test procedures to support the evolved design?
6. What is the appropriate level of resource commitment for enabling technologies?
 - An effective Agile T&E process depends on the quality of test automation and configuration management that is available and applied. The availability of a high-fidelity staging environment promotes more rapid and effective implementation potential.
7. How does the PM evaluate development progress and predict program development schedule/budgets?

- The team needs to maintain priorities and minimize work in progress; what should the FAA’s oversight process look like?
8. What is the role of the FAA’s Solutions Delivery Team during Sprint and Release development (including integration and T&E)?
 9. What are effective configuration management tools and techniques?
 - As there may be parallel development and test efforts and environments, the software baselines need to be integrated on a continuous basis (e.g., nightly) to ensure that the developers and testers are working on the same baseline.
 10. How does the PM ensure that the team is comfortable raising issues in a team setting, and promote a collaborative environment to address any impediments that are identified during the retrospective?
 11. How do the PMs on both the FAA and contractor sides elevate and address impediments that involve the working relationship between the two groups?

3.3 Deployment

3.3.1 Purpose

The purpose of deployment is to make developed capabilities available for operational use. More specifically, the deployment should promote, if not ensure, that provided capabilities reach and are used by the intended end users. Agile deployment assumes a context in which user acceptance tests that meet the acceptance criteria have been completed, the software release to be deployed has been configured for the operational environment, and necessary enablers such as training, operating instructions, and other support are provided.

3.3.2 Deployment Factors and Practices

The approach adopted for an Agile deployment depends on a number factors:

- Change capacity of the operational environment
- The operational community’s demand for change
- Overhead of Release preparation and deployment
- Complexity (e.g., numbers sites/positions/installations, dependencies on and compatibilities with interfacing systems)
- Extent of test and deployment automation.

Although some of these factors are inherent to the situation, potential issues can be mitigated by observing a number of best practices:

- Automate testing to the extent possible.
- Employ a staging environment as equivalent as possible to the operational environment.
- Document as needed.

3.3.3 Deployment Pipeline

Each development and deployment will reflect the processes employed by the developer, the specific aspects of the system/capability being developed, and the transition procedure between development and operations. Figure 3-3, depicted in Section 3.2.2, presents a generic T&E and deployment pipeline.

When a capability or package of capabilities is determined to be ready for operation, it is subject to operationally oriented tests conducted with/by the user community. When accepted by the representative testers of the user community, the new operational baseline is ready to be staged for release to operations. Figure 3-4, depicted in Section 3.2.2, describes an example for release acceptance criteria.

Early and frequent user feedback is a fundamental feature of Agile development. At the completion of each Sprint, users have the opportunity to examine and respond to completed functions, ideally in the form of demonstrations of integrated capabilities. Deployment of operational capability to the users represents the final stage of this user feedback process, enabling user feedback on the integrated product in the operational environment.

In addition to subjective user feedback, the usefulness of delivered capabilities can be assessed through measurements of feature use and post-operational analysis of effectiveness in achieving operational objectives. System analysis recording capabilities and offline data analysis can inform users and developers about the frequency and impact of feature use, and user surveys can provide subjective feedback of user perceptions. Consistent with this view, the Acquisition Management System promotes post-implementation reviews that are intended to verify the extent to which operational objectives and asserted benefits are realized.

During deployment, the pace of trouble reports may increase and observed problems may be more site-unique. It is important that site personnel be actively involved in the backlog management process and that the process be responsive to sites' priorities.

Although Agile values working software over documentation, effective training is fundamental. Training may involve an approach such as user and operator manuals, but might also include more technology-oriented approaches such as computer-based training, help features with new capabilities, or training simulations that enable users and operators to gain familiarity with capabilities in offline or non-operational modes of the system. An Agile approach to training should involve solicitation and response to user feedback regarding training content, and iterative development of training materials responsive to user needs.

3.3.4 Roles and Responsibilities

Table 3-13 captures the roles and responsibilities for the FAA and contractor during deployment.

Table 3-13. Roles and Responsibilities for Deployment

Role	Responsibility
FAA - Business Owner	Responsible for informing user community of planned operational changes.
FAA - PM	Responsible for planning, scheduling, coordinating, and allocating resources to meet deployment commitments.
FAA - Product Owner	Supports the PM and helps coordinate and manage the feedback from the operations
Contractor -Development Team	Supports release configuration and deployment logistics, as appropriate.
Users/Stakeholders	Responsible for product familiarization, training conduct, transition support, and product feedback.

3.3.5 Deployment Challenges

Whether Agile or not, deployment of systems or new capabilities to existing systems is always challenging. Deployment challenges unique to Agile relate to the Agile objectives of providing

deployable capabilities earlier and more frequently than traditional development approaches. Each Agile deployment situation is different because of the varying needs of the operational environment and user community. Each deployment situation is influenced by the resources committed to enabling mechanisms that support more frequent deployment. With these considerations in mind, specific challenges of Agile deployment include:

1. What is the best deployment cadence?
 - Determination of deployment cadence involves constraints represented by the operational environment (i.e., the ability of operations to absorb changes), the development cadence adopted (i.e., the duration of Sprint and Release cycles), the granularity of Features being developed and implemented, and the maturity/stability of Agile process execution.
2. Do you have quality and available supporting enablers?
 - The Agile guidance to provide “just enough” documentation always represents a challenge. For deployment, determining what is enough involves operating instructions, training documentation, updated system design and description, and other supporting documentation required to meet operational needs.
3. Have you determined the division of responsibility between developer and operational community?
 - The Agile objective of more frequent delivery of deployable capabilities suggests a greater degree of integration between developer and operational activities. Determining how and when that transition occurs depends on the operational situation and the contracted scope of development activities.
 - Further discussion with appropriate stakeholders needed to clarify the role of the FAA’s Product Owner during Deployment

3.4 Special Topics

This section addresses the following special Agile program management topics:

- Contract Management
- Cost Estimation
- Agile Metrics.

3.4.1 Contract Management

Applying Agile effectively in a contract requires considerable planning and coordination between the contracting organization, the program, and eventually the performing contractor. As COs are responsible for obligating government funds, they serve as the bridge between the FAA and its industry partners, where contract execution requires a close partnership with the Program Management Team to conduct an Agile acquisition. Each program/project should have a COR to assist in contract management. For an Agile project, the COR role may be served by someone from the contracting office who is working on the program (specifically with the PM and Product Owner) or may be an added responsibility for the Product Owner. A COR is the CO’s eyes and ears for the technical performance of the contract. According to the 2012 version of the FAA COR Handbook [13], the COR’s primary responsibilities include finalizing the government requirements, monitoring the contractor’s performance, and inspecting and accepting the delivered product.

Key practices for structuring the contract include the following:

- Arrange the contract around an outcome-based approach.
- Ensure that the contract supports frequent delivery of capability.

- Ensure that the contract supports performance metrics in order to evaluate contractor’s performance (accountability).

An outcome-based Agile contract may use either a Performance Work Statement or a Statement of Objectives, since either one enables the contractor to construct the solution based on the FAA’s desired mission outcomes rather than detailing the aspects of how to perform the work. One approach to implementing Agile is to follow the concept of modular development as described by the Office of Management and Budget [14]. While this document contains multiple references to parts of the Federal Acquisition Regulations from which the FAA is exempt, it can easily be adapted to FAA use. The value of modular contracting is that it is designed to realize usable capabilities in one or more successive, interoperable increments.

The use of Indefinite-Delivery, Indefinite-Quantity or Blanket Purchase Agreement vehicles provides contractual flexibility that can enable Agile practices. These vehicles allow the FAA to issue small task orders to multiple vendors for short periods of time without having to go through the whole process of awarding stand-alone contracts to a single vendor. These orders could be structured to allow for a quantity of individual Sprints of a given size to be ordered at one time (e.g., six one-month duration Sprints) or as a “Release of (TBD) months duration containing (TBD) Sprints.” Once the acquisition approach is chosen, the team must determine the appropriate type of contract. Other federal agencies often prefer Firm Fixed Price (FFP) contracts, since the contractor is usually required to deliver a predetermined item (e.g., a product) before payment is made. With FFP, it may be possible to fix the duration of the Sprint and the resources, allowing the requirements baseline to flex. Here, the requirement for payment would for the contractor to deliver the entire level of effort.

An alternative approach would be to use Cost-type and Time and Materials contracts to acquire Agile functionality within a set cost threshold, which the contractor could not exceed. Both of these contract types allow flexibility in scheduling and in the incremental delivery of functionality. However, the usage of these two types of contracts requires greater government oversight to monitor progress, and to ensure that the contractor delivers a usable product within the required timeframe and estimated cost.

Table 3-14 captures the roles and responsibilities needed to support the development of the solicitation.

Table 3-14. Roles and Responsibilities for Contract Management

Role	Responsibility
FAA - PM	<p>During procurement planning and contract formation:</p> <ul style="list-style-type: none"> • Coordinates with the CO to ensure that the solicitation addresses Agile considerations (may include outlining expectations in SOW regarding Agile metrics and or Agile software development metrics). • Considers how the contractor’s proposed effort supports the attainment of the Project Vision and meets the objective of the program roadmap. Clearly defines the expected outcome in the contract. • Understands the contractor’s approach to managing and implementing an Agile effort. <p>During contract administration: Coordinates with CO/COR regarding the scope of the contract with respect to the program status (related to contract modifications).</p>

Role	Responsibility
FAA - CO/COR	<p>During procurement planning and contract formation:</p> <ul style="list-style-type: none"> • Ensures that the solicitation clearly addresses Agile considerations. • Clearly defines with the FAA PM the expected outcome in the contract and the necessary Agile metrics to evaluate the contractor’s progress. • Establishes evaluation criteria to ensure that the FAA has sufficient oversight. The contract may include the following: <ul style="list-style-type: none"> ○ Description of contractor’s Agile development expertise. ○ Proposed staffing approach and experience of the Development Team (includes technical background of lead personnel, but does not prescribe any certifications or qualifications). ○ Past performance on Agile program of similar size, scope, and complexity (may have to address private sector performance in addition to government contracts). The focus should be on how well the contractor has implemented a solution utilizing its proposed Agile process. ○ Description of how the contractor will manage, execute, and measure its Agile development approach to meet end user objectives and maintain quality (may also be necessary to specify FAA oversight, such as access to backlog). <p>During contract administration:</p> <ul style="list-style-type: none"> • Assists contractor in derivation of technical requirements (CO is authorized to provide technical direction and must work with FAA PM to ensure alignment with contract). • Recommends and makes changes to contract, if necessary. <p>Inspects and accepts any contractual deliverables.</p>

There are some challenges associated with contract management, which must be evaluated prior to execution. All parties involved in the solicitation, procurement planning, and contract administration need to have a common understanding of Agile techniques to ensure that the proper language and intent is captured correctly. The CO and COR assigned to a program must be able to assist the PM in making business decisions and trade-offs that come with implementing an Agile acquisition and development process, without impeding the cadence of developing and delivering working capabilities. For example, the PM and CO need to balance the requirement for certain documents/artifacts from the contractor with the Agile value of having working software over comprehensive documentation. Agile promotes the idea that these development life-cycle artifacts are living, and the PM and CO must be cognizant of that. While the COR is the CO’s technical liaison, special consideration must be made to clearly delineate the responsibilities between the COR and the Product Owner, regarding providing technical guidance to the Development Team. The COR must collaborate with the Product Owner, if identified as two distinct roles within the FAA Program Management Team, to ensure the alignment of technical guidance and contractual commitments.

3.4.2 Estimation

An Agile program, just like any other program, needs to have an understanding of what the cost of performing the required effort will be over the program’s life. Agile estimation is based on the concept of

“just enough”; therefore, an Agile estimate has a higher level of uncertainty at the Epic level, and is defined by the Program Team. However, at the Sprint level, the estimate has a greater degree of certainty, since it is developed for a short, well-defined period by the team performing the work.

A program should use a traditional top-down approach where the Product Owner, Lead Engineers, and PM use historical cost and performance data to answer the two basic questions relating to estimation: 1) What are the tasks? and 2) How long will they take? Since detailed planning has not been completed during program initiation, the project timeline and cost accuracy are not very precise and should be expressed as a range. This estimate should allow the PM to identify the level of funding required for the program at the highest level and then allocate the funds across the program. During the Program and Release Planning stages, the Program Team will develop progressively more detailed estimations and resource allocations with the contractor Development Team’s input. At lower levels (e.g., Sprint Planning and development), these estimates become part of the program budget request, meaning that PMs of an Agile program must be able to adapt to changes in the work plan in order to meet the funding availability.

Agile estimates are typically based on comparisons of the work to be performed with work completed in the past. Some of the common Agile estimation techniques, in order of degree of accuracy, are:

- Relative Comparison – evaluating whether a particular User Story is more complex/larger than another (e.g., A is bigger than B and B is bigger than C).
- Historical – evaluating the complexity of a User Story based on previous similar efforts.
- Story Points – evaluating the relative complexity of a User Story expressed in points (Fibonacci Series: 1, 3, 5, 8 ...) or size (small, medium, large...). It is an abstract method for evaluating the activity at a high level.

3.4.3 Agile Metrics

Throughout development, the PM must monitor and assess the status of the contractor’s work for each Sprint and Release. Agile promotes several metrics that may be used to assess the Development Team’s progress, including burn-down, defects backlog (quality), coefficient of variation for team performance predictability, deferred complexity (risk), burn-up, and velocity. These metrics are based on estimations defined in Section 3.4.2, calculated by the Development Team, and are continuously refined to assess the progress of the program in meeting its objectives [15] ([16] provides more information regarding executive dashboard categories for measures and metrics for oversight).

During the Sprint, the contractor Development Team should produce these performance metrics: the burn-down chart, the burn-up chart, and velocity. A burn-down chart, as shown in Figure 3-5, is a graphical presentation of development team’s progress measured in story points for a Sprint. The blue dotted-line represents an ideal straight-line burn-down. The solid orange line with punctuated data points represents actual work accomplished each day. If the team always finishes early, there may be a mismatch in the team’s ability and its commitments (e.g., the team is under-committed and is able to do more). Conversely, if the team cannot finish the work within the Sprint time-box, the team may be over-committed. If the chart shows a steep decline instead of gradual burn-down, the team may not have planned the effort with enough granularity. More specifically, the “Team Behind” and “Team Ahead” labels, in Figure 3-5, identify instances during the sprint where the Development Team is not performing consistently and at risk of not meeting the Sprint commitments by the end of the Sprint. A Release Burn-down chart would be very similar to the Sprint chart, where the x-axis would instead represent the number of Sprints in a Release versus days in a Sprint.



Figure 3-5. Sprint Burn-down Chart

The Sprint burn-up chart, in Figure 3-6, depicts delivery of backlog items measured in story points over a number of Sprints. The blue line represents the delivery of story points over time (what is remaining). The orange with data points line is the actual delivery during the Sprint. The dotted orange line is a forecast of how long it will take to complete at present delivery level (the target level). A burn-up chart will show the impact of “scope creep” (see blue solid line), or the addition of new requirements into a project after the initial requirements had been defined. While tolerating scope creep during a Sprint is bad practice, scope change within a Release is expected with Agile development. As the team moves through the project, the Product Owner may decide to take on or remove work based on a refined understanding of the business need. From the PM perspective, this chart provides information regarding the health of the program, enabling the PM to evaluate issues because of certain decisions made during a Sprint (e.g., scope creep), and information regarding the contractor’s performance. With this information, the PM may evaluate if the Development Team cannot develop a potentially shippable product or make progress toward the Sprint goals. A Release Burn-up chart would be very similar to the Sprint burn-up chart, where the x-axis would represent each iteration instead of each day within a Sprint.



Figure 3-6. Sprint Burn-up Chart

The defect burn-up chart, in Figure 3-7, captures the output from testing and how the Development Team is addressing it. Bugs/defects are captured throughout T&E during a Sprint. They are managed by either fixing them within the Sprint to claim success with respect to a User Story, or keeping defects and User Stories separate from each other, and addressing issues in parallel with development efforts. The latter is dependent on how the team is set up and their development level. For example, if the Development Team includes testers working along with the software developers, then defects may be addressed at the unit and integration levels. Or if there is a separate Quality Assurance Team (or Test Team), then bug fixing may be done in parallel with the development effort. The monitoring of defects is essential for the FAA and contractor PMs, as it provides information if the PMs need to take corrective action. In Figure 3-7, the PMs may assess the cumulative count of defects (blue line) against those that have been addressed (orange line). This graph highlights that there are more bugs identified than the team is able to address.

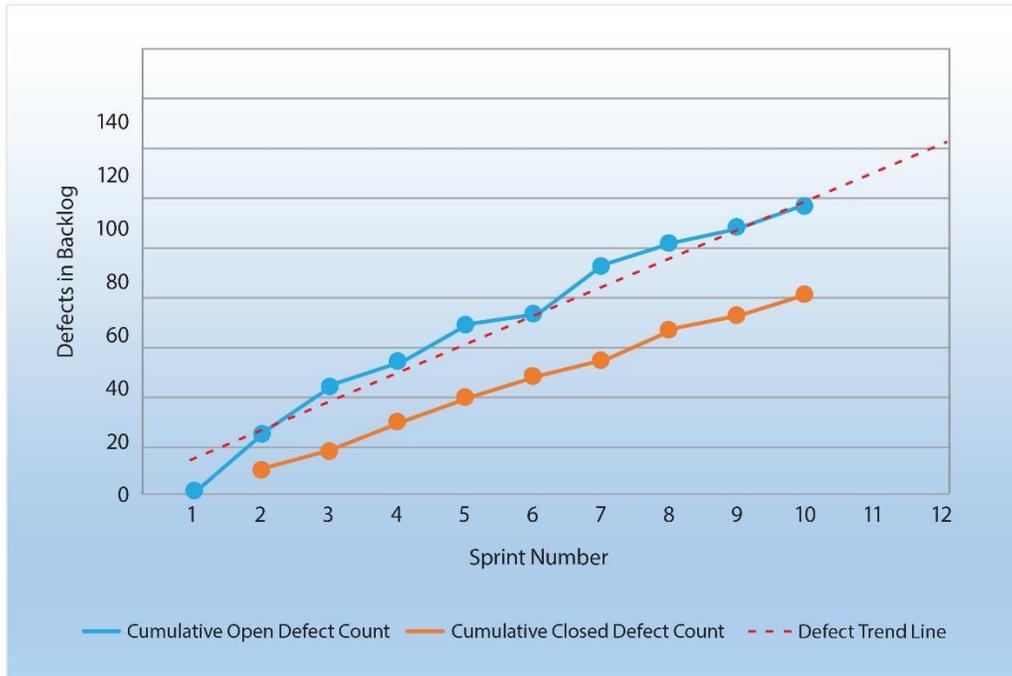


Figure 3-7. Defect Burn-up Chart

Velocity, as shown in Figure 3-8, is the amount of work a team accomplished in a specified period of time, measured in story points. It is the basis for forecasting the team’s ability to complete the work planned. Since velocity evolves over time and applies to a specific team, it may be an unreliable indicator of future performance if the composition of the Agile team or the technology changes frequently. New teams can expect to see an increase in velocity over time as the team optimizes relationships and the work process. A decrease in average velocity is usually a sign to the team and the PM that some part of the team’s development process has become inefficient.

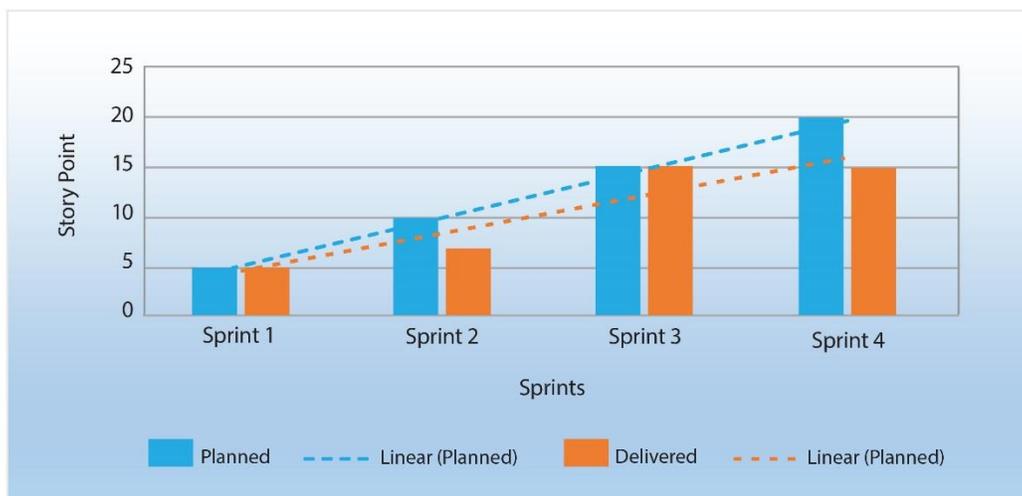


Figure 3-8. Velocity

4 Summary and Additional Considerations

This document provides Agile program management guidance for FAA PMs. It identifies best practices related to planning, development, deployment, and other program management responsibilities, enabling the benefits from employing Agile methodologies in a program.

4.1 Agile PM Summary

Table 4-1 shows some of the considerations that can be expected when managing an Agile program. The left-hand column is the topic area, the center column is a high-level statement of the considerations, and the right-hand column is the section in this report where a more detailed discussion of the topic can be found.

Table 4-1. Summary of Agile Program Management Considerations

Considerations in Agile Program Management		
Topic	Management	Section
Planning	<ul style="list-style-type: none"> • Ensure that adequate Agile planning expertise is available. • Define the “minimum viable product.” • Ensure collaboration among the Program Management Team, Business Owner, and Development Team. 	3.1
Development	<ul style="list-style-type: none"> • Define when a Release is “done.” • Balance need between documentation and software. • Ensure that “part-time” Product Owners meet responsibilities. • Evaluate development progress. • Ensure design coherence and integrity. • Manage complexities of integration testing and system of systems testing. • Predict schedules and budgets. • Define the FAA Solutions Delivery Team roles. • Establish a configuration management process. • Ensure a comfortable and collaborative environment. • Effectively elevate and resolve inter-group issues. 	3.2
Deployment	<ul style="list-style-type: none"> • Determine the best deployment cadence. • Determine the level of resource commitment for enabling technologies. • Ensure quality and availability of supporting enablers. • Determine the division of responsibility between developer and operations communities. 	3.3
Contract Management	<ul style="list-style-type: none"> • Ensure that all parties have a common understanding of Agile techniques. • Ensure that required artifacts from the contractor are of value and do not become more important than working software. • Ensure that the CO/COR assists in making business decisions and trade-offs without impeding the cadence of delivery. • Delineate responsibilities between COR and Product Owner, and ensure collaboration. 	3.4.1

Considerations in Agile Program Management		
Estimation	<ul style="list-style-type: none"> • Ensure that the level of estimate certainty increases as the Agile process is executed. • Adapt work plan changes to meet funding availability. 	3.4.2
Metrics	<ul style="list-style-type: none"> • Maximize team workload without over-commitment. • Minimize scope creep. • Maximize team velocity. 	3.4.3

4.2 Agile PM Additional Information

For additional information regarding Agile program management, please see the following:

- For FAA Agile Program Management guidance, please see Section 3.8 in “Federal Aviation Administration Agile Acquisition Principles and Practices” [2]
- For PMI Certification information, please see PMI-ACP (Agile Certified Practitioner) [17]
- For estimation guidance, please see GAO *Cost Estimation and Assessment Guide* by GAO [18]
- For Earned Value Management guidance, please see *Earned Value Management Application Guide* by NDIA [19]
- For Agile Acquisition guidance for federal government, please see *Best Practices to Procure Agile IT Services* by ACT-IAC [20].

5 References

- [1] Federal Aviation Administration, "Program Management Practice Toolkit," August 2016. [Online]. Available: https://ksn2.faa.gov/faq/AcquisitionProfessions/Practices/Pages/Disc_PM.aspx.
- [2] Federal Aviation Administration and The MITRE Corporation, "Federal Aviation Administration Agile Acquisition Principles and Practices," April 2016. [Online]. Available: <http://fast.faa.gov/docs/FAA AgileGuide.pdf>. [Accessed August 2016].
- [3] Beck, K., et. al., "Manifesto for Agile Software Development," 2001. [Online]. Available: <http://agilemanifesto.org/>. [Accessed August 2016].
- [4] Beck, K., et. al., "Principles behind the Agile Manifesto," 2001. [Online]. Available: <http://www.agilemanifesto.org/principles.html>. [Accessed August 2016].
- [5] R. Peter Measey, Agile Foundations Principles, practices and frameworks, BCS, 2015.
- [6] M. Cohn, "Scrum," Mountain Goat Software, [Online]. Available: <https://www.mountaingoatsoftware.com/agile/scrum>. [Accessed August 2016].
- [7] K. S. & J. Sutherland, "Scrum Guides," July 2016. [Online]. Available: <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf#zoom=100>.
- [8] U.S. Digital Service, "TechFAR Handbook for Procuring Digital Services Using Agile Processes," 7 August 2014. [Online]. Available: https://playbook.cio.gov/assets/TechFAR%20Handbook_2014-08-07.pdf. [Accessed August 2016].
- [9] United States Government Accountability Office, "Agencies Need to Establish and Implement Incremental Development Policies GAO-14-361," May 2014. [Online]. Available: <http://www.gao.gov/assets/670/662922.pdf>. [Accessed September 2016].
- [10] Scaled Agile, Inc., "Team Demo Abstract," Scaled Agile, Inc., 15 October 2015. [Online]. Available: <http://www.scaledagileframework.com/team-demo/>. [Accessed August 2016].
- [11] Scaled Agile, Inc., "Solution Demo Abstract," Scaled Agile, Inc, 31 March 2016. [Online]. Available: <http://www.scaledagileframework.com/solution-demo/>. [Accessed August 2016].
- [12] Scaled Agile, Inc., "System Demo Abstract," Scaled Agile, Inc. , 21 April 2016. [Online]. Available: <http://www.scaledagileframework.com/system-demo/>. [Accessed August 2016].
- [13] Federal Aviation Administration, "Contracting Officer's Representative Handbook," 2012. [Online]. Available: <http://fast.faa.gov/docs/CORHandbook.doc>. [Accessed August 2016].
- [14] Office of Management and Budget, "Contracting Guidance to Support Modular Development," 14 June 2012. [Online]. Available:

- <https://www.whitehouse.gov/sites/default/files/omb/procurement/guidance/modular-approaches-for-information-technology.pdf>. [Accessed August 2016].
- [15] W. Hayes, S. Miller, M. A. Lapham, E. Wrubel and T. Chick, "Agile Metrics: Progress Monitoring of Agile Contractors," Carnegie Mellon University, Hanscom AFB., MA, 2014.
- [16] S. Miller, W. Hayes, E. Wrubel and SEI Agile in Government Team, "Written Testimony of Software Engineering Institute's Agile in Government Team to House Ways and Means SSA Subcommittee," 14 July 2016. [Online]. Available: <http://waysandmeans.house.gov/wp-content/uploads/2016/07/20160714SS-Testimony-Hayes.pdf>. [Accessed August 2016].
- [17] Project Management Institute , "PMI Agile Certified Practitioner (PMI-ACP)," PMI, 2014. [Online]. Available: <https://www.pmi.org/certifications/types/agile-acp>. [Accessed August 2016].
- [18] United States Government Accountability Office, "GAO Cost Estimating and Assessment Guide GAO-09-3SP," March 2009. [Online]. Available: <http://www.acqnotes.com/Attachments/GAO%20Cost%20Estimating%20Guide%202009.pdf>. [Accessed August 2016].
- [19] National Defense Industrial Association, "Earned Value Management Systems Application Guide Revision 2," 27 January 2015. [Online]. Available: http://www.ndia.org/Divisions/Divisions/IPMD/Documents/NDIA_IPMD_Application_Guide_Rev_Jan272015.pdf. [Accessed August 2016].
- [20] ACT-IAC, "Acquisition Best Practices to Procure Agile IT Services," March 2014. [Online]. Available: <https://www.actiac.org/system/files/Best%20Practices%20to%20Procure%20Agile%20IT%20Services%20-%20ET%20SIG%202003-2014.pdf>. [Accessed August 2016].

Appendix A List of Abbreviations

Acronym	Definition
CO	Contracting Officer
COR	Contracting Officer's Representative
FAA	Federal Aviation Administration
FAR	Federal Acquisition Regulations
FFP	Firm Fixed Price
ISPD	Implementation Strategy Planning Document
IT	Information Technology
OT&E	Operational Test and Evaluation
PM	Program Managers
SSD	System Specification Document
SOW	Statement of Work
T&E	Test and Evaluation
TEMP	Test and Evaluation Master Plan
V&V	Verification and Validation
XP	Extreme Programming

Disclaimer

The contents of this material reflect the views of the author and/or the Director of the Center for Advanced Aviation System Development (CAASD), and do not necessarily reflect the views of the Federal Aviation Administration (FAA) or the Department of Transportation (DOT). Neither the FAA nor the DOT makes any warranty or guarantee, or promise, expressed or implied, concerning the content or accuracy of the views expressed herein.

This is the copyright work of The MITRE Corporation and was produced for the U.S. Government under Contract Number DTFAWA-10-C-00080 and is subject to Federal Aviation Administration Acquisition Management System Clause 3.5-13, Rights in Data-General, Alt. III and Alt. IV (Oct. 1996). No other use other than that granted to the U.S. Government, or to those acting on behalf of the U.S. Government, under that Clause is authorized without the express written permission of The MITRE Corporation. For further information, please contact The MITRE Corporation, Contract Office, 7515 Colshire Drive, McLean, VA 22102 (703) 983-6000.

©2016 The MITRE Corporation. The Government retains a nonexclusive, royalty-free right to publish or reproduce this document, or to allow others to do so, for “Government Purposes Only.”



MITRE