



A Framework for Applying Agile Methods for Non-Information Technology (IT) Disciplines

**Authors: Nadya Huleatt
Kurt Miller
David Yu
William Brice**

The views, opinions and/or findings contained in this report are those of The MITRE Corporation and should not be construed as an official government position, policy, or decision, unless designated by other documentation.

©2022 The MITRE Corporation.
All rights reserved.

McLean, VA

**Approved for Public Release;
Distribution Unlimited. Public Release
Case Number 22-0142**

This page is intentionally blank.

Executive Summary

Today's organizations are facing a complex landscape of issues, whether building an open collaborative work environment or addressing questions arising from technological adoption, such as automated process, cloud computing, and artificial intelligence. To respond to this rapidly changing environment, additional capabilities, processes, and tools are needed to develop solutions that provide value to impacted stakeholders and align with intended outcomes.

In recent years, there has been a growing number of organizations looking at innovative methods for developing non-Information Technology (IT) products, such as policy. Organizations are evaluating how adopting these innovative methods address the political, technical, and economical risks to deliver value to their customers. One innovative method is Agile, wherein cross-functional teams develop requirements and solutions while receiving rapid, iterative stakeholder feedback. Agile is a proven software development method, successfully used by industry and government, to address the risk of failing to deliver value to products in a timely and cost-efficient manner. In turn, organizations in non-IT disciplines are exploring how Agile values and principles provide an iterative, accountable, and focused development and deployment method for addressing stakeholder needs and delivering value, given resource and time constraints.

This paper shows how Agile, specifically a modified Scrum framework, may be applied in a non-traditional IT context. This Agile practice implements a disciplined process that addresses stakeholder needs and ensures the desired outcome is understood and achieved. Building upon past experiences, we present an execution process and apply it toward an organizational policy scenario to demonstrate a structured, incremental, and iterative policy development approach. Our intended audience is organizations who are unfamiliar or new to Agile and interested in learning how to apply Agile principles in non-IT related disciplines.

This page is intentionally blank.

Table of Contents

1	Introduction.....	1
1.1	Agile and Scrum Overview	1
1.2	Purpose and Outline	3
2	Policy Framework and Agile Values	3
3	Scenario for Applying Agile for Policy Development	4
4	Proposed Agile Development Process	5
4.1	Culture and Organizational Structure.....	5
4.2	Vision	8
4.3	Planning for Development and Product Roadmap	9
4.4	Epics.....	12
4.4.1	Operationalizing Policy through Architecture.....	13
4.4.2	Minimum Viable Product (MVP).....	14
4.5	Product Backlog.....	15
4.5.1	Features	16
4.5.2	User Stories	17
4.5.3	Estimation.....	19
4.6	Sprint.....	20
4.6.1	Sprint Planning.....	20
4.6.2	Development	21
4.6.3	Standup.....	22
4.6.4	Backlog Refinement.....	23
4.6.5	Review.....	24
4.6.6	Retrospective	26
4.7	Measures	27
4.7.1	Product Measures	27
4.7.2	Team Measures.....	27
5	Operationalizing Agile Development	28
6	Conclusion	30
7	References.....	31

List of Figures

Figure 1. Scrum Flow [6]	2
Figure 2. Policy Development Model	3
Figure 3. Product Vision [6].....	8
Figure 4. Agile Planning Onion [6].....	9
Figure 5. Product Roadmap [6]	10
Figure 6. Product Roadmap for Portfolio Management	11
Figure 7. Populating the Product Backlog [6].....	15
Figure 8. Planning, Standup, Product Development, Review, and Retrospective [6].....	20
Figure 9. Backlog Refinement Session Agenda [6]	24

List of Tables

Table 1. Agile Development Team Roles and Responsibilities	5
Table 2. Future Workforce Ecosystem Project Agile Policy Team	7
Table 3. Future Workforce Ecosystem Project Stakeholders	7
Table 4. Future Workforce Ecosystem Project Epics.....	12
Table 5. Future Workforce Ecosystem Project Features	16
Table 6. Future Workforce Ecosystem Project User Story and Acceptance Criteria	18
Table 7. Future Workforce Ecosystem Project Definition of Done	22
Table 8. Agile Transition Roadmap	29

1 Introduction

Today's decision makers are facing a complex landscape of issues. For example, building an open collaborative work environment or addressing questions arising from advancements in technology, such as artificial intelligence and machine learning, requires additional tools in their organization's toolbox. To respond to this rapidly changing environment, additional capabilities and processes are needed to develop solutions that provide value to stakeholders and align with intended outcomes.

A growing number of organizations are looking at innovative methods for developing their non-IT related products. This paper focuses on the Agile method, wherein cross-functional teams develop requirements and solutions while receiving rapid, iterative stakeholder feedback. Government agencies and industry have evaluated Agile methods, tailored their practices around it, and successfully demonstrated the value of applying Agile for software intensive programs in various domains, such as Agile software acquisitions for aviation [1], or manufacturing and project management [2]. Agile has also evolved organizational processes—such as acquisition and systems engineering—to accommodate iterative and incremental development methods.

In the following sections, we explore how government agencies and industry may adopt these Agile values, principles, and methods, using a tailored Scrum framework, to address the development and implementation risks in non-IT disciplines. We build upon our experience in applying Scrum to various government, non-IT related challenges and use a policy development example to demonstrate its application.

In this paper, we propose an execution process for developing non-IT solutions in an incremental and iterative manner, which continue to test and evolve. This execution process is applied in a theoretical policy scenario that is common to all organizations and agnostic to a specific domain and respective policies. Our intention is to provide implementation guidance and examples of application and lessons learned that demonstrate how a disciplined, Agile process may ensure stakeholder's policy needs are addressed, and the desired policy outcome is understood and achieved.

1.1 Agile and Scrum Overview

The Agile Manifesto [3] is the foundation of Agile methodology and espouses the development and deployment values oriented around working products, collaboration, and responding to change. Execution methods rooted in Agile include Kanban, Scrum, and Scaled Agile Framework to name a few, and are often applied to the management, development, and deployment of software solutions [4]. Scrum is an Agile framework developed by Jeff Sutherland and Ken Schwaber and used to complete complex projects worldwide since the early 1990s [5]. This paper specifically focuses on Scrum when referencing Agile and applying it to developing nontechnology-centric solutions. Based on our experiences, we tailored and modified Scrum to include additional elements that are not in the baseline process in The Scrum Guide [5], but provide a more structured work management method to:

- Deliver highest possible value through early and regular coordination with stakeholders,
- Focus on team commitments for product delivery versus individual development,
- Establish a predictable development cadence, and
- Promote visibility into the development status.

The Scrum cadence includes the following events to create regularity (Figure 1):

Scrum Flow

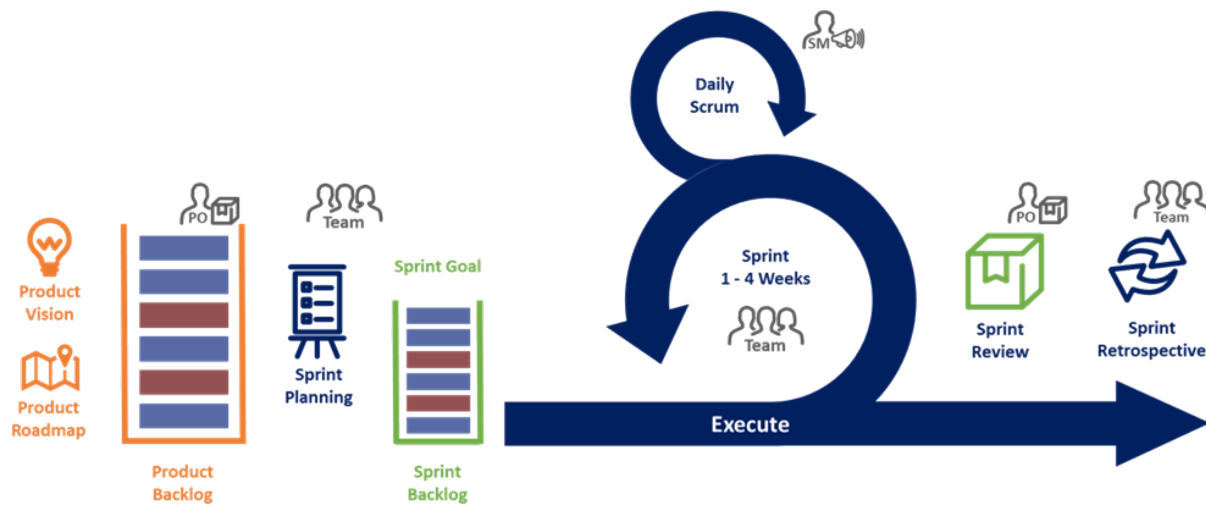


Figure 1. Scrum Flow [6]

- **Sprint:** 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 based on the Agile Development Team’s capability and the complexity of the product’s intent.
- **Sprint Planning:** This event is held to elaborate, estimate, and prioritize highest-value product backlog items, create the Sprint backlog, and agree on Sprint Goals.
- **Daily Scrum (or Standup):** The Daily Scrum is a 15-minute event where the Agile Development Team discusses progress toward the Sprint Goal(s). The intent is to share and determine progress for a given Sprint and to identify obstacles.
- **Sprint Review:** Reviews provide an opportunity for the entire team, including stakeholders (e.g., user community) and product developers, to come together and evaluate what they accomplished during the Sprint—relative to the Sprint plan and Sprint backlog. The developed set of stories are demonstrated, and feedback is used for subsequent planning events.
- **Sprint Retrospective:** This final event in the Scrum process is an important piece of the “continuous improvement” mind-set that provides an opportunity to reflect upon the processes, tools, and team organization. Issues and mitigation strategies are placed in the product backlog and prioritized for implementation in successive Sprints.

1.2 Purpose and Outline

This paper is intended to serve as an execution process for applying Scrum to develop non-traditional applications, specifically for those who are new to Agile or who wish to better understand the use of Agile in non-IT disciplines. We focus on the policy development process to demonstrate the value and benefits of applying Agile principles and methodologies. The perceived benefits of using Agile for policy development include the following:

1. Increase the quality of impactful policy products by delivering small changes frequently
2. Quickly iterate to obtain clarity and stakeholder consensus on developing policy that can be implemented, while achieving intended outcomes
3. Continuously assess progress and priorities and balance them with the team's ability to incrementally deliver outcomes

This paper includes:

- Section 2 – presentation of a general policy development framework
- Section 3 – policy scenario that serves as examples for applying Agile
- Section 4 – proposed Agile process for non-IT applications, with specific practice examples around policy development and lessons learned for implementation
- Section 5 – proposed roadmap for operationalizing Agile within a policy organization.
- Section 6 – concludes with reminders of Agile principles that will enable success when using Scrum to develop nontechnical solutions, like policy development.

2 Policy Framework and Agile Values

A policy is a specified system of principles that guides decisions and moves individuals and organizations to achieve rational outcomes. The Centers for Disease Control and Prevention specifies policy as “law, regulation, procedure, administration action, incentive, or voluntary practice of governments and other institutions” [7]. In this paper, policy includes the logical argument that is captured and implemented through a document or rule.

While no one singular, general theory of policy development exists, models focus on three key concepts: 1) identifying issues; 2) developing and implementing solutions; and 3) monitoring and evaluating policy effectiveness [8], [9]. Identifying the issue, the desired outcome, and evaluation measures are fundamental to policy development, and qualify what defines “good policy.”

Sound policy development should also be defined by a policy vision. The policy vision sets the future view of the policy and provides the purpose of what the policy seeks to achieve. The policy vision should reflect user and stakeholder needs and must clearly articulate the intent, change needed, and expected actions that one needs to take. As the model in Figure 2 shows, the policy vision surrounds the policy development process. Policies should be developed and evaluated in alignment with the policy vision to achieve desired outcomes.

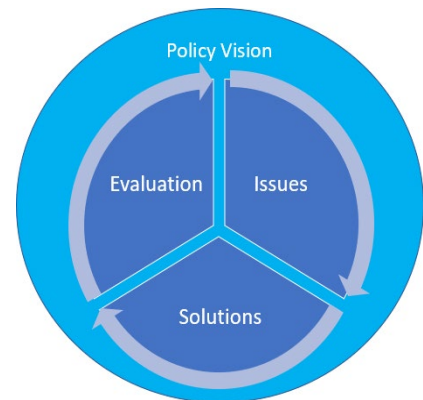


Figure 2. Policy Development Model

General themes of effective or “good” policy development emphasize upfront planning to evaluate the issue(s), determine priorities, and develop policy solutions. Clearly understanding the issues within the current environment is critical. Key factors to consider are the political, economic, social, technological, and legal environments [10]. Addressing policy issues requires data on precisely what needs to be fixed and having clear objectives among stakeholders and policymakers: foundational questions address who the stakeholders are; what needs to change; how much it will cost; how long it is expected to take; and what the potential, unintended consequences may be.

At its core, policy should be formed with implementation in mind [11, pp. 538-560], therefore the upfront planning should include how the implementation will be evaluated. Policy effectiveness cannot be evaluated without a predetermined set of well-defined metrics by which the success of the policy can be measured and then used to determine if additional adjustments are required in the future. All policies should include mechanisms for monitoring and evaluation to determine the policy’s effectiveness [12]. Monitoring and evaluation play an important role before, during, and after implementation.

3 Scenario for Applying Agile for Policy Development

As we move forward to describe the use of Scrum in a non-IT environment, we present a scenario that focuses on a small team within a fictitious organization making workforce changes and capturing those changes in policy. This scenario anchors implementation examples presented throughout the document, demonstrating how a cross-functional team can use an iterative development approach to develop and operationalize sound policy.

In today’s globally connected environment, organizational restructuring, defined by McKinney and Scherer as “any major reconfiguration of internal administrative structure that is associated with an intentional management change program, [13, pp. 735-752]” has become a common event. Whether it is a major overhaul of the organization’s business model or a minor change to the organization’s structure, most authors recommend developing a policy that explains the why, what, and how. Because of the number of organizational adjustments in recent years by organizations of all size, this scenario seems like a logical example to which most readers can relate. It is not intended to describe how to restructure an organization, instead, it is merely a means to illustrate the use of Scrum in a non-IT situation to develop policy.

In our scenario, almost two years of a global pandemic caused a mid-sized global organization to transition the majority of their workforce to work remotely—a significant change from their pre-pandemic, on-site business model. The leadership anticipates future disruptions and external forces that will require new and updated organizational policy to address the unique and collaboration needs of their mixed on-site, hybrid, and remote workforce. Once the pandemic showed signs of receding, the organization’s leadership established the Future Workforce Ecosystem Project and stood up a cross-functional Agile Policy Team to evaluate their past on-site and current remote work model and use an iterative approach to develop both a plan and a policy for the organization’s future work model.

Over the past few weeks, the Agile Policy Team collected data to guide their policy development. Workplace surveys sent to the entire organization highlighted that a portion of the organization felt disconnected from the company and their coworkers, and the current remote working resources did not adequately support their virtual collaboration needs. In recent communications, the organization’s leader stated “... (our) organizational brand includes ‘Culture as a competitive edge and people (are) critical to culture.’ ” The Agile Policy Team reviewed organizational documentation on culture and found that the organization values collaboration as a cultural attribute stating, “We recognize that at the

intersection of inclusion and diversity is innovation, and therefore we strive to collaborate across our diverse population, projects, and teams to unlock our collective potential.”

The Vice President overseeing the Future Workforce Ecosystem Project provided guidance that going back to having a physical presence is no longer a firm requirement for the organization. And in her recent townhall style meetings across the organization, she learned the workforce values the flexibility to work remotely.

4 Proposed Agile Development Process

This section provides the details of an Agile development process that could be used to produce non-IT products, provide examples of applying Agile for developing policy, recommended implementation notes, best practices, and lessons learned where appropriate.

4.1 Culture and Organizational Structure

Agile principles, rooted in Scrum, drive the need for a different team structure with modified roles and responsibilities, daily accountability, and milestones to ensure feedback is continuously captured and addressed, and that a valuable product is developed and released in a timely manner. Table 1 identifies members of the Agile Development Team: Product Owner, Scrum Master, and development team members. These teams should be built around motivated individuals that leadership, stakeholders, and individual team members trust and support to produce the desired product (policy and policy documentation). To succeed, these individuals must work together daily, effectively collaborating to build and shape the intended product.

Table 1. Agile Development Team Roles and Responsibilities

Role	Responsibility
Product Owner	<ul style="list-style-type: none">• Authorized to determine the vision of the product.• Maximizes the value of the work conducted by the Agile Development Team and serves as a motivator.• Interfaces with interacting stakeholders.• Manages the product backlog as the authorized representative of the user community (manages stakeholder expectations). The Product Owner may reprioritize, remove, or update items in the backlog as necessary (For policy, this may include external influences from industry or Congress).• Approves and accepts User Stories, provides Acceptance Criteria and the definition of “done,” and provides technical direction to the team during the Sprints.
Development Team	<ul style="list-style-type: none">• Maximizes the value of the work through incremental development of the product.• Self-organizing, cross-functional team that may consist of policy writers, Information and Technology Engineers, organizational representatives, and policy Subject Matter Experts (SMEs).• Responsible for executing the Sprint and participates in planning and development related activities.• Empowered to make decisions and able to work at a sustainable pace, adjusting its processes and behaviors as necessary.

Role	Responsibility
Scrum Master	<ul style="list-style-type: none"> • Acts as servant leader for the Agile Development Team. • Enforces the rules of Scrum and serves as the point-person for resolving team-based and organizational impediments. • Coaches the team without dictating, ensuring there is a safe and sustainable working environment.

The following are additional, key roles that are not identified within the Scrum framework, but who work with the Scrum Team throughout the development:

- **Users:** Users are individuals who will be impacted and will benefit from the product. Users provide feedback during the Agile Development Process to define the required value of the solution. For policy, this may include individuals who would have a role in enacting and adhering to the policy.
- **Stakeholder Community:** A stakeholder represents others who have an interest or responsibility for the product. Stakeholders include those who identify and prioritize changes in the product (such as feedback or policy deviations) or are responsible for approval. Depending on the individual role and level of influence, they may provide feedback when appropriate or serve on the Agile Development Team, if appropriate (e.g., a SME). Design Thinking tools, to understand and empathize with the users, like Stakeholder Analysis should be leveraged to understand who is involved, identify their needs, and understand which influential stakeholder needs to be directly involved in the creation/update and approval.

The use of the Design Thinking activity of developing Personas can be useful when the Agile Development Team does not have direct access to the entire target user community. A Persona is a descriptive model of a person (user, stakeholder, team member, etc.); this tool is most often used to help a team define and understand the needs of its customer. (See MITRE's Innovation Toolkit for example templates for conducting stakeholder analysis. [14], [15])

If the future product impacts a large user population or requires the input of a large community with vested interest in the solution, it is appropriate to identify and engage with a user or stakeholder representative while developing the product. As an example, representatives for an organization's employees could be the human resources department, employee community of interest groups, or trade unions. When the solution impacts those outside the span of control of the organization responsible for development, the Agile Development Team may use the Stakeholder Analysis to identify relevant and important organizations that represent these broader communities of interest (e.g., government agencies, national or international professional associations, non-business entities, unions). The use of user or stakeholder representatives adds an additional layer of complexity during planning, but it will not change the Agile development process.

The following best practices support the culture, team, and organizational shift toward Agile:

- Ensure users and key stakeholders understand that they will have a more proactive role in shaping the product to address the need and provide input, feedback, and approval, as appropriate, throughout the lifecycle.
- Use fully dedicated resources to the extent possible, especially when standing up an Agile Development Team.
- Effectively integrate the members of the product team, composed of those with various expertise (e.g., technical SME, policy writers, technical editors).

- Build a team environment that is collaborative and trusting.
- Instantiate self-organizing, cross-functional, empowered teams and allow them to operate with relative autonomy.
- Complete a Stakeholder Analysis and define a change management plan to assist with the cultural and organizational adjustments.
- Ensure the Product Owner understands and commits to the upfront work, such as crafting the Product Vision, before work can reasonably begin. This initial work is critical for providing development direction.
- When establishing and forming an Agile Development Team, carefully distinguish between selecting individuals who are highly influential stakeholders that will provide feedback and individuals with the required skills who should be part of the Agile Development Team. Members of the Agile Development Team should be individuals that you expect to do work.

Future Workforce Ecosystem Project Scenario Example

The organization selected personnel that represented their core business areas to be part of the Future Workforce Ecosystem Project's Agile Policy Team. This team, shown in Table 2, is responsible for developing the appropriate workforce policy changes. The first action the Agile Policy Team took upon their formation was to conduct a stakeholder analysis to best understand their organizational environment and who were the important stakeholders. Table 3 shows which stakeholders will be expected to provide feedback to shape the new policy and their areas of influence.

Table 2. Future Workforce Ecosystem Project Agile Policy Team

Team Members	Agile Policy Team Role
Mission Lines of Business Representatives	Agile Policy Team Member
Organizational Change Subject Matter Expert	Agile Policy Team Member
Enterprise Information and Technology Representative	Agile Policy Team Member
Workforce of the Future Collaboration Project Lead	Product Owner
Talent Acquisition, Diversity, and Inclusion Representative(s)	Agile Policy Team Member

Table 3. Future Workforce Ecosystem Project Stakeholders

Stakeholder	Stakeholder Influence
Chief Executive Officer	Responsible
Finance	Support
Mission Lines of Business Directors	Support
Department Representatives (including Managers)	Support
Legal Representative	Support
People and Corporate Strategy Vice President	Accountable

Lessons Learned: When implementing Agile at the team level, organizational and leadership buy-in for Agile is key to successfully ensuring that an organization can keep up with today's accelerated pace of change; lack of understanding and communication among leadership and the workforce may inhibit adoption of this fundamental shift in the operating model and organizational culture.

4.2 Vision

The Product Vision should take an aspirational, achievable, realistic view of the future; looking at the future solution, how it will operate in the future environment, and how will this future be achieved. The Product Vision should:

- Reflect user and stakeholder needs (e.g., congressional mandates, industry changes, public outcry);
- Provide a mission statement for the product; and
- Describe the boundaries and constraints that influence the product and decisions to be made.

The Product Owner will shape the Product Vision, shown in Figure 3, to define what the product will accomplish and its associated business value. The Product Owner will coordinate with key stakeholders to frame the Product Vision.

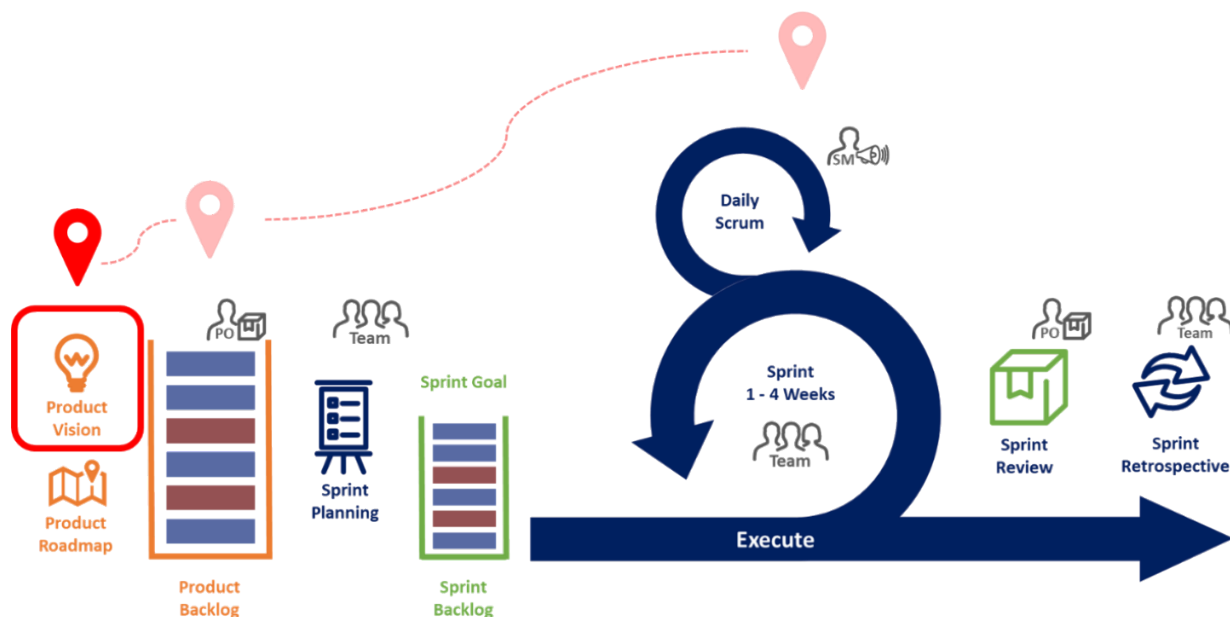


Figure 3. Product Vision [6]

Sources of the Product Vision include operational shortfalls, organizational strategies, priorities, and business plans. The output of a Stakeholder Analysis will identify those stakeholders who will have greater influence in shaping the Product Vision. Conduct a problem-framing exercise to help identify needs and opportunities from a user's perspective. Align Epics and the Product Vision with problem-framing statements so that the Agile Development Team understands the purpose and what “done” will look like.

Future Workforce Ecosystem Project Scenario Example

Effective policy should not have gaps or create ambiguity. Policy must clearly articulate the intent of the change, the need for change, and required actions. A policy Product Vision is a future view of the policy product and provides the larger, contextual overview and value of the policy under development.

To evaluate the workforce ecosystem of the future, the Agile Policy Team put together the following aspirational vision statement:

The future workforce ecosystem provides the ability to produce innovative solutions to our customers most difficult problems in a flexible and collaborative environment.

Lessons Learned: *The Product Vision is not a static statement of the future product, and the Product Owner must continuously evaluate the validity of the Product Vision, updating the vision to reflect changes in environment and/or needs, as appropriate, and coordinating with the stakeholders as appropriate.*

4.3 Planning for Development and Product Roadmap

An Agile mindset emphasizes continuously adjusting plans based on lessons learned. An Agile project conducts planning at multiple levels, as shown in Figure 4, based on various planning horizons. The focus is on delivering product(s) that will help the user achieve business value.

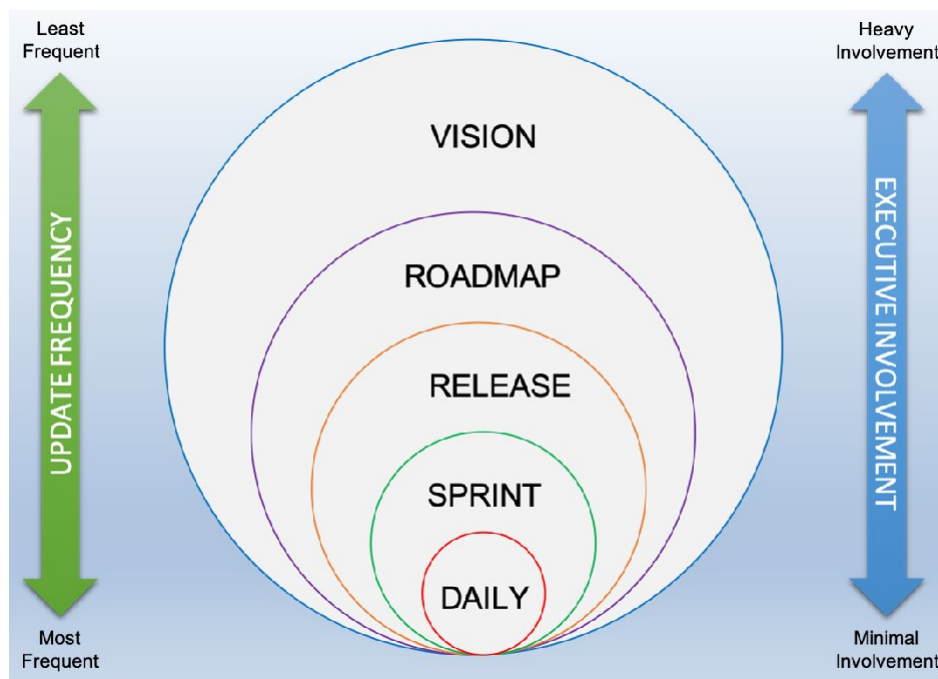


Figure 4. Agile Planning Onion [6]

The layered approach toward planning ensures that the right stakeholders will be engaged at the right level of detail throughout the process. This is consistent with the Agile value of responding to change over following a plan:

- For long-term planning: Strategic planning is captured in the Product Vision and Product Roadmap. It is like traditional project planning, as the Product Vision articulates the purpose and context for the effort, and the Product Roadmap outlines how the product will be developed and evolved over a significant period of time (over 6 months and longer) and what resources will be needed.
- For mid-term planning: Release planning is both strategic and tactical in nature. The Release represents the most highly valued Features that have been designed, developed, and evaluated

through a series of Sprints, and is ready for delivery every 2-3 months. The Release plan focuses on (1) what will be delivered every release, and (2) on defining and developing an MVP that may be defined around evaluating assumptions for policy and evolving workforce procedures. A Release plan may be developed to capture and track which Features are planned, turning the Product Roadmap into a detailed actionable plan. Release planning occurs at a regular cadence, which is commensurate with the duration of Release development.

- For near-term planning: Sprint planning is tactical in nature. It has a shorter planning horizon and orients the work around high-priority stories and tasks that can be accomplished within a fixed timeframe (typically 2–4 weeks).

The Product Roadmap (see Figure 5) describes the planned evolution of the product based on strategic priority. Specifically, it identifies the prioritized Epics to be developed and delivered during the implementation timeframe. The roadmap provides a sense of direction and lays out a realistic plan to fulfill the Product Vision and achieve the Minimum Viable Product (MVP) and additional capabilities beyond the MVP, if required. It helps with communicating purpose, alignment of expectations, and prioritization; however, it should not be considered a fixed plan or commitment. The roadmap shares the implementation plans, status of effort, and priorities; the specific details of these items may vary based on the intended audience, that is, more details may be appropriate for the Agile Development Team to guide development efforts and fewer details are necessary when communicating milestones to the users and executive leadership.

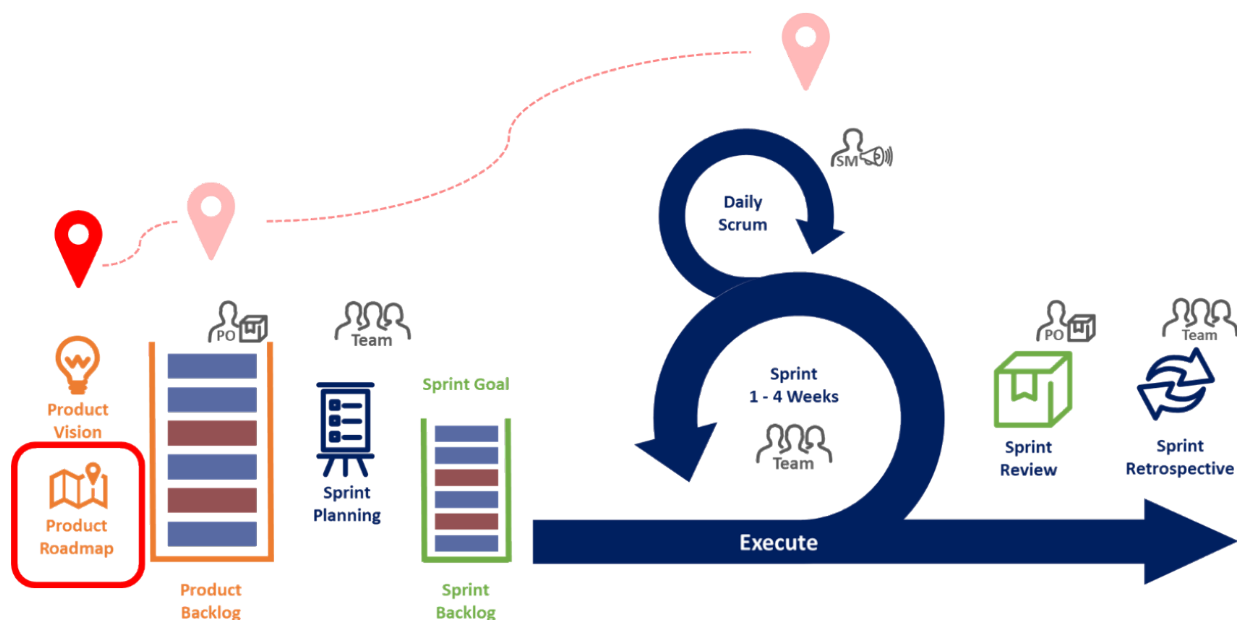


Figure 5. Product Roadmap [6]

The Product Owner may wish to hold a formal Roadmap Session (meeting) to develop the Product Roadmap and invite relevant stakeholders, including the Agile Development Team and Scrum Master, to provide feedback and inputs into shaping the Product Roadmap. This 2–4 hour Roadmap Session event would occur at the beginning of a project as part of overall planning and as Epics/Features are being defined and the product backlog is populated. The Product Owner may hold additional Roadmap Sessions as necessary to update the roadmap based on changes to the product or additional stakeholder feedback. The goal of this event is to identify and agree upon outcomes of effort, expectations for success, and identify potential roadblocks.

When developing a Product Roadmap, consider how to address the following questions:

- Why are we building the product?
- What are we hoping to accomplish?
- How will this help users?
- Where have we seen these before?

By answering these questions with the relevant stakeholders, the Agile Development Team can focus on developing what is most important, which becomes the product's baseline. Teams may consider various approaches for addressing these questions and use tools like Jira, Microsoft (MS) PowerPoint, MS Excel, or Trello to draw out what the roadmap may look like:

- Top down (sequential order based on a process)
- Themes or Epic-based
- Storyboard a visual road
- Bottom Up (build from items identified in the backlog – least preferred method).

Regardless of the method used, it is important to remember that the Product Roadmap is not fixed, and it should be continuously revisited and revised as product development plans evolve, adhering to the Agile principle for welcoming changes even late in development.

Figure 6 is an example Product Roadmap, highlighting the Epics that will be completed per quarter and key milestones.

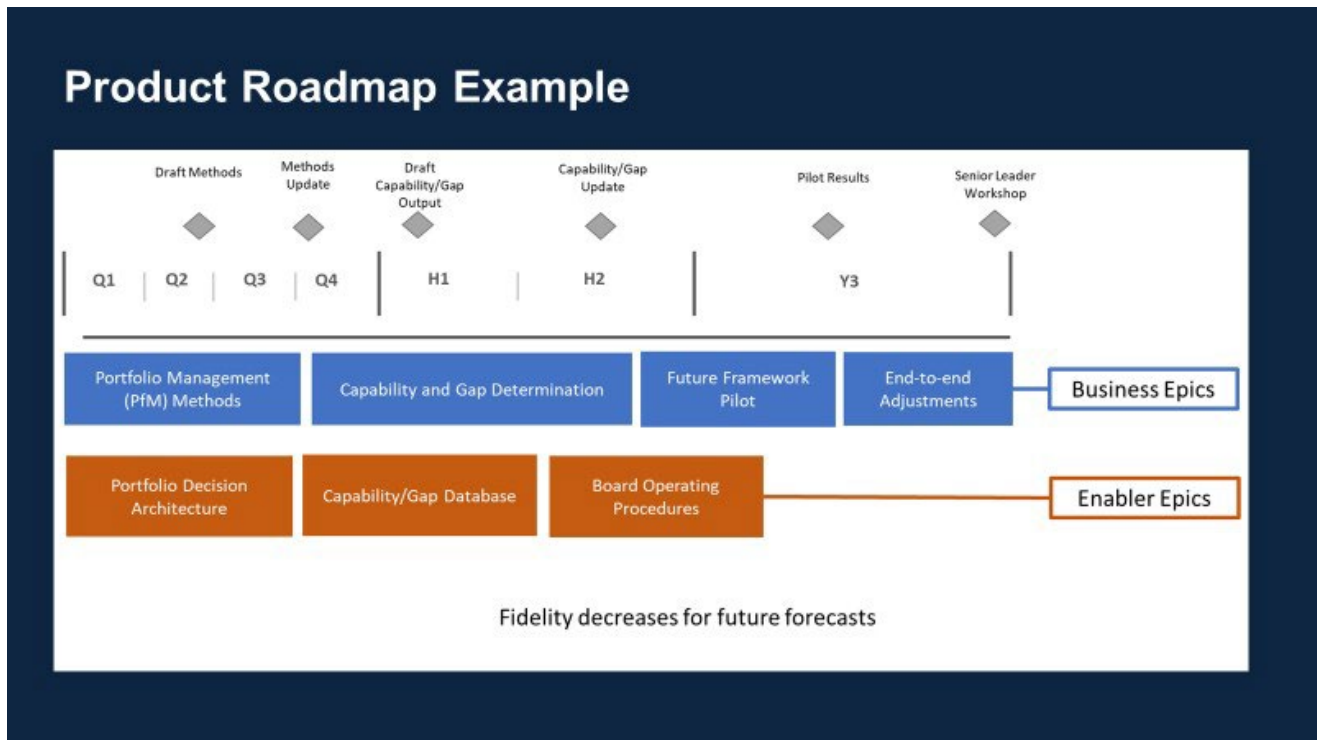


Figure 6. Product Roadmap for Portfolio Management

4.4 Epics

Broadly, an Epic represents a large body of work that may be broken into Features, where the work may be related to developing pieces of functionality or identifying enablers, such as requirements for a solution tool. Epics typically span multiple releases. An Epic communicates the business needs and intent for the desired product, like policy. Epics can cover the required procedural topics or changes within policy, such as cybersecurity and changes in procedures related to risk analysis. Epics are managed in a backlog, arranged into realizable, prioritized groupings. The Product Owner will work with the users to elicit, identify, and prioritize the Epic(s) and its value proposition, and will be responsible for maintaining the product backlog.

Example format for an Epic [16]:

- Short Title
- Description
 - For (target user)
 - Who (statement of the need or opportunity)
 - The (product name) is a (product category)
 - That (key benefit)
 - Unlike (primary alternative)

Future Workforce Ecosystem Project Scenario Example

From the Product Vision, the Product Owner produced the following Epics (See Table 4) to provide the necessary guardrails for the solutions space without intentionally providing prescriptive direction that may be constraining.

Table 4. Future Workforce Ecosystem Project Epics

Number	Epic	
E1	Epic Title	Develop an organizational view of the past and current workforce ecosystem
	Epic Description	For the organizational workforce who is responsible for delivering business value the organizational view is a description of the current and past state of the organization that we can use to identify opportunities for improvements unlike others who have implemented workforce ecosystem changes without understanding the ‘as-is’ state
E2	Epic Title	Develop workforce ecosystem opportunities
	Epic Description	For the organizational workforce who is responsible for delivering business value the workforce ecosystem is a workforce environment that is more resilient unlike current workforce ecosystem that is not adaptable to disruptors that significantly impact the workforce value creation

Lessons Learned: *The Product Vision influences how Epics are defined, and Epics are intended to guide exploration and development efforts, where the team may conduct exploratory activities to address uncertainty and to refine their thinking during ideation.*

4.4.1 Operationalizing Policy through Architecture

Architecture for policy broadly represents the intentional formation/structure of the policy documentation. For this paper, we propose two considerations for architecture. First is the organizational environment a policy is developed within, and second is the structure of the policy document.

Merriam-Webster [17] gives two definitions of architecture that are applicable to policy development:

2. *A: formation or construction resulting from or as if from a conscious act //*
the architecture of the garden

B: a unifying or coherent form or structure //a novel that lacks architecture

Definition 2.a. offers an understanding for the first level (“formation or construction resulting from or as if from a conscious act”), where policy is developed within an organization’s process and rules for policy formulation. Definition 2.b. (“a unifying or coherent form or structure”) provides an understanding for the policy document [17].

An Agile architecture must:

- Evolve over time while supporting the needs of current users,
- Balance emergent design and intentionality, and
- Provide the necessary structure so that the “system” always runs.

The current policy environment could be based on a standard waterfall methodology that requires a linear set of steps from policy update identification to policy documentation approval. The Product Owner using an Agile approach may need to coordinate the approach with the policy owning organization to gain permission to pilot a new approach.

There may be organizational guidance that specifies how documentation must be structured, including specifying minimum sections in a generic template, while still providing flexibility. The architecture may vary across documents, aligning with the Agile principle of allowing the best architectures, requirements, and design to emerge from self-organizing teams and allowing teams to determine the best method to communicate requirements and changes to the users. An example of how documentation may be architected embraces including sections of policy that address the following topics:

- To clarify the reasons for change
- To describe the benefits of the change
- To offer a detailed picture of the new organization or organizational actions
- To describe how the change will take place, including the timeframe for change and/or coordination needs with bargaining units
- To provide information on support/resources.

One consideration is the idea of modular architecture, where a section of the policy documentation is specifically scoped so it may be updated without significantly impacting or causing rework in other sections of policy.

Lessons Learned: *Architecture should be driven by the Product Owner or even a Policy Architect, who is responsible for determining how requirements in the policy must be organized, such as procedural appendices, or in templates or attachments. It is important that the team include a User Story, at a minimum, around establishing the architecture, and that the Product Owner should prioritize this User Story in initial Sprints.*

4.4.2 Minimum Viable Product (MVP)

Traditionally, teams drive for completed or near complete products/policy prior to seeking stakeholder input, which often causes schedule delays as the team works on addressing all feedback and incorporating changes, even those with low or no value. An MVP is an Agile concept intended to reduce the chance of providing products with little or no value. An MVP is an early version of a developed product for users or user representatives to evaluate and provide feedback. User and stakeholder feedback on the MVP helps determine if the capability meets the user's needs or requires future enhancements and design. There are three possible outcomes when developing an MVP.

1. The MVP is completed, determined to provide sufficient user value, and no further development is required.
2. The MVP is completed, determined to need additional value, and further development is continued to meet the user needs. Iterative development continues until the product contains the appropriate amount of value to meet user needs.
3. The MVP is completed, it is determined the expected result does not justify the additional effort, and development is stopped.

Defining an MVP follows the Agile principle around the value of simplicity and maximizing the amount of work not done, and provides the following benefits:

- Prevents waste building something with little value to the stakeholders
- Gets the product into the hands of users quickly, enabling fast feedback
- Quickly proves or disproves your hypothesis.

A prioritization matrix that maps out urgency and impact is one method for determining the MVP. The concept of an MVP should be discussed often, especially when there are conflicting priorities that need to be worked on with a limited amount of time and resources: priority should be given to items that deliver highest user impact with lowest levels of effort from the team.

Future Workforce Ecosystem Project Scenario Example

In this scenario, the Agile Policy Team worked with a representative set of stakeholders comprising department representatives, including employees and managers, to seek user needs for the future workforce ecosystem solution. Based on the Agile Policy Team's understanding of the workforce ecosystem of the future requirements, the Agile Policy Team developed an MVP description:

In the event of a national health emergency, a future workforce ecosystem with descriptions that enumerate the need for change, the solution, and policy statements that specify what is changing procedurally and organizationally.

To realize the MVP, the Agile Policy Team followed a Sprint cadence (described in Section 4.6) to develop aspects of the policy; solicit, capture, and prioritize feedback; and address comments. While not all voices may be heard through a representative user sample, seeking their feedback about the MVP early on may help identify and address issues, thus preventing future implementation delays by understanding what is most valuable to the workforce.

4.5 Product Backlog

The product backlog is an ordered list of work for the Agile Development Team, which can be reorganized to reflect changes in priorities. The product backlog includes requirements that are organized based on

- User priority,
- Urgency of getting feedback,
- Relative implementation difficulty, and
- Symbiotic relationship between work items.

The backlog is generally populated with Epics, Features, and User Stories that operationalize the Product Vision. The product backlog, as shown in Figure 7, captures the Epics, Features and User Stories. Backlog Refinement is an event where the Product Owner will review and prioritize items in the product backlog with key stakeholders to ensure alignment. The Product Owner should have enough complete User Stories in the backlog to support 2-3 Sprints before work on a new product begins.

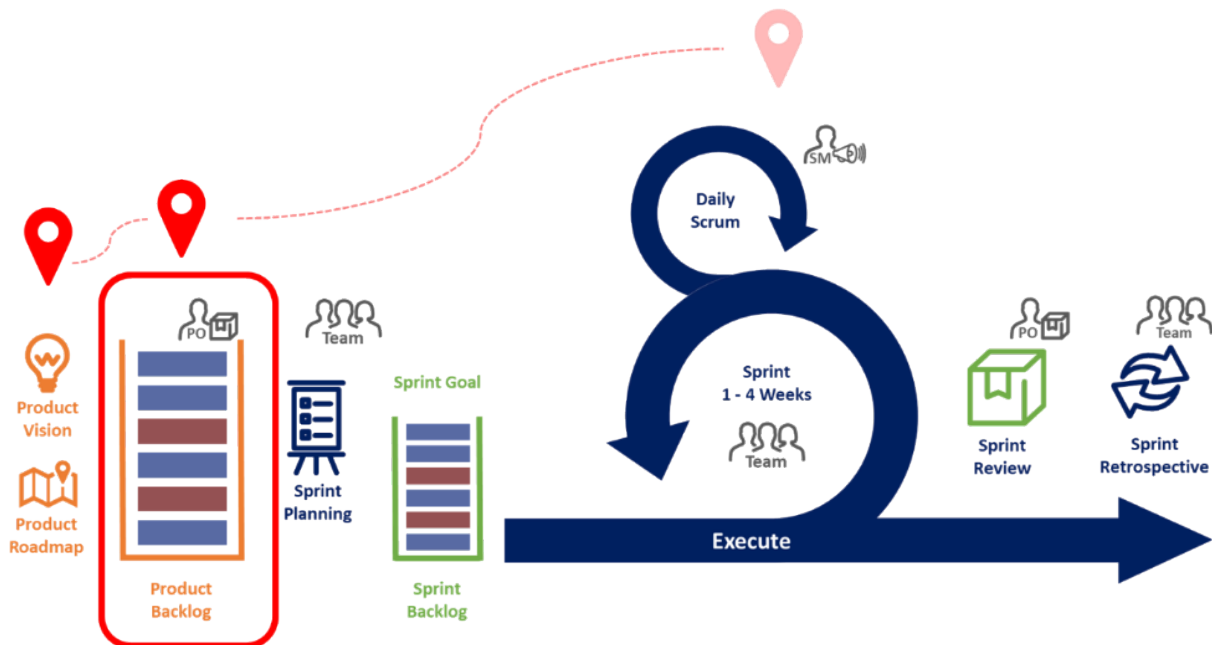


Figure 7. Populating the Product Backlog [6]

Lessons Learned: Some potential issues to watch out for regarding the backlog include the following:

- The Product Owner does not adjust the backlog based on feedback from the Agile Development Team and stakeholders.
- The Agile Development Team limits items in the backlog to only items that are user-centric and do not account for non-functional/enabler stories that are necessary when developing product.
- The backlog is locally stored and shared infrequently, meaning the team and/or stakeholders are unable to view the status.

4.5.1 Features

A Feature describes functionality that delivers business value provided by the product to fulfill stakeholder needs and address product's intent. Features are not typically part of the baseline Scrum process and have been specifically included in this Agile process for non-IT to decompose the MVP into greater granularity for planning purposes. Features deliver new value in a single release. Typically, multiple Features are derived from an Epic (a many-to-one traceability). The Feature may represent aspects of the requirements, applicability, and accountability that need to be well documented for a given operational need. Features are generated to build an understanding and consensus between the Agile Development Team, users, and stakeholders.

Acceptance Criteria defines the Feature's boundaries and is used to define a Feature as done. Acceptance Criteria helps the Product Owner answer what is needed for a Feature to provide value and helps the Agile Development Team gain a shared understanding.

Example of a Feature format:

- Short Title
- Description
- Acceptance Criteria

Future Workforce Ecosystem Project Scenario Example

Deriving from the Epics, the Product Owner collaborated with other Future Workforce Ecosystem Project stakeholders and the Agile Policy Team to identify relevant Features, as shown in Table 5, to understand the best workforce environment that addresses the many needs and challenges their organization is facing. In an organizational policy context, Features define the guardrails around purpose, accountability, responsibilities, application, and action (procedures), and identify areas of flexibility and innovation for the future workforce.

Table 5. Future Workforce Ecosystem Project Features

Number	Feature	
E1	Epic	Develop an organizational view of the past and current workforce ecosystem
E1.F1	Feature Description	Short Title: Current Organizational Description Description: As-is workforce description of the structure and enablers Acceptance Criteria: detailed organizational structure (current and past 5 years)
E2	Epic	Develop workforce ecosystem opportunities
E2.F1	Feature Description	Short Title: Innovation Processes Description: Organizational capabilities/functions to support adoption of innovative processes and products Acceptance Criteria: Increase workforce satisfaction by 10%

4.5.2 User Stories

A User Story represents desired functionality, expressed in a concise, simple description from a user's perspective, and is captured and managed as part of the Sprint backlog. The Product Owner and Agile Development Team members may develop User Stories with the following characteristics:

- Expressed in the user's business language
- Captures who, what, and why
- Represents a small piece of value that be realized in a Sprint (this may be functional or an enabler/support function)
- Includes criteria to determine when the User Story is complete (Acceptance Criteria)
- Can be prioritized and collected in a Sprint backlog owned by the Agile Development Team
- Completed in a single Sprint

One method for defining a User Story is based on the idea of vertical slices, which represent end-to-end functionality that is valuable to the user. The stories should represent what the user will see and experience, and not specify the solution in the User Story itself, as the Sprint is intended to allow the Agile Development Team to explore the solution space.

A User Story consists of the following:

- Short title
- User Story description (As a <user> I want <goal> so that <desired benefit>.)
- Acceptance Criteria
- Acceptance Tests
- Notes/Assumptions/Diagrams
- Dependencies

If there is too much ambiguity within a User Story, the Product Owner and Agile Development Team may write research-related or proof of concept stories to investigate the alternatives and solicit feedback for delving into a more complex solution.

Acceptance Criteria is part of the User Story description and specifies what a completed User Story looks like (i.e., when a User Story has been satisfied). It sets the scope, helps the developer know when to stop adding functionality and defines the users' expectation for a single User Story that the Agile Development Team needs to meet. Acceptance criteria also helps the Agile team derive the appropriate tests.

Future Workforce Ecosystem Project Scenario Example

Table 6 shows User Story generated by the Product Owner for the future workforce environment policy.

Table 6. Future Workforce Ecosystem Project User Story and Acceptance Criteria

Number	User Story and Acceptance Criteria	
E1	Epic Title	Develop an organizational view of the past and current workforce ecosystem
E1.F1	Feature	Current Organizational Description
E1.F1.US1	User Story Description	<p>User Story 1 Short Title: identify the current workforce skills, knowledge, and abilities Description: As a Future Workforce Ecosystem Policy Writer, I want to identify the current workforce knowledge, skills, and abilities (KSAs), so that I understand the current workforce competencies Acceptance Criteria: List of top 15 KSAs from the past 5 years divided into Public and Private business sectors Acceptance Tests: Peer review by Public and Private Sector Human Resources SMEs Notes/Assumptions/Diagrams: KSA data provided by Human Resources stored in folder KSA on the shared drive Dependencies: None</p>
E1.F1.US2	User Story Description	<p>User Story 2 Short Title: describe the current organizational structure and operating rules Description: As a Future Workforce Ecosystem Policy Writer, I want to identify the program division's organizational structure, so that I understand the interrelationships and connections among teams and departments Acceptance Criteria: Description of how departments within the division are connected Acceptance Tests: Peer review by Public and Private Sector Human Resources SMEs Notes/Assumptions/Diagrams: None Dependencies: None</p>
E2	Epic Title	Develop workforce ecosystem opportunities
E2.F1	Feature	Innovation Processes

Number	User Story and Acceptance Criteria	
E2.F1.US1	User Story Description	<p>User Story 1</p> <p>Short Title: Identify face-to-face collaboration capabilities</p> <p>Description: As a Future Workforce Ecosystem Policy Writer, I want to identify collaboration capabilities so that I advance face-to-face teamwork at the two main campuses.</p> <p>Acceptance Criteria: Capabilities must be easily available and accessible to the workforce; capabilities must secure the information exchanged internally; capabilities must allow visual and auditory sharing of information for physically co-located staff</p> <p>Acceptance Tests: Peer review of methodology and output provided</p> <p>Assumptions: Initial list is not constrained by cost</p> <p>Dependencies: Availability of audio/visual SMEs</p>

Lessons Learned: Guard against having a solution in mind when writing User Stories as it may bias the User Story and constrain the value you deliver. Adhere to the Agile principle of allowing architecture, requirements, and design to emerge from self-organizing teams.

4.5.3 Estimation

Estimation is fundamental to product measures and team metrics. It is a method to determine what may be realistically accomplished within a fixed timeframe (e.g., Sprint), and may be used by the Agile Development Team to drive consensus as they plan work.

Abstract, relative measures of work is the foundation for estimating Epics, Features, and User Stories. Methods such as T-shirt sizing (e.g., small, medium, large) or story pointing may be used to provide a rough estimate of the time or level of effort needed to complete an Epic, Feature, or User Story. For example, an “extra small T-shirt” estimation for an Epic means that the Epic may be completed within a single release and an “extra-large” means that the Epic is highly complex and requires significant effort across multiple releases. Story points, with a Fibonacci series, is the most common method for User Story estimation, where a single number represents an individual’s perspective on the following:

- Volume: How much is there?
- Complexity: How hard is it?
- Knowledge: What do we know?
- Uncertainty: What don’t we know?

These User Story points are also based on team consensus and are neither influenced by who will ultimately be working on a User Story, nor by capacity (availability) in a Sprint. Teammates with lower capacity in a specific Sprint will simply take on fewer User Story points. There are several recommended ways to begin estimating to build team consensus around the baseline and meaning of User Story points, including starting with the simplest User Story (e.g., User Story point of 1), or starting with an average sized User Story as a reference point from which the team can decide what stories are bigger and smaller. This is a team exercise where it is recommended that everyone anonymously give their own estimates before everyone simultaneously reveal their estimates at the same time to the group. Other than serving as a way for team members to better estimate capacity for each Sprint, the exercise of pointing stories will also help gain consensus on User Story expectations and give the Product Owner one last opportunity to add details to the Acceptance Criteria, especially when there are multiple individuals with huge discrepancies of estimations.

Lessons Learned: A common pitfall is to not elicit feedback from both junior and senior team members. Ensuring there is a ‘team’ consensus of a story’s relative size avoids over or under estimation.

4.6 Sprint

Figure 8 depicts the several events that happen during every Sprint: Planning, Standup (Daily Scrum), Product Development, Review, and Retrospective.

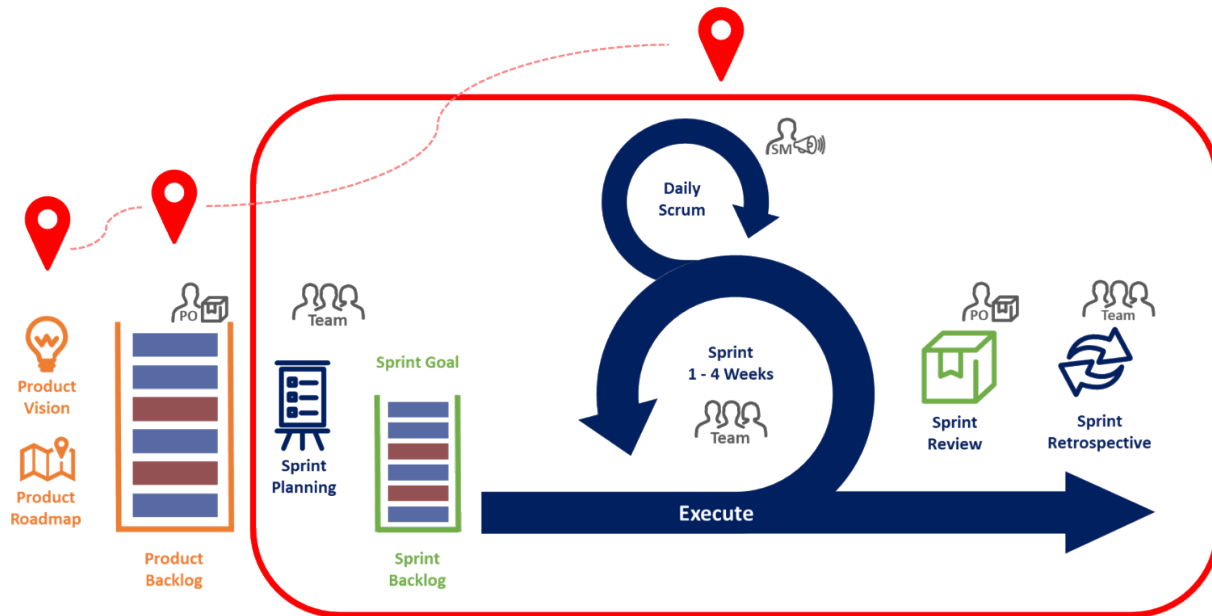


Figure 8. Planning, Standup, Product Development, Review, and Retrospective [6]

4.6.1 Sprint Planning

Sprint Planning is a Scrum event where the Product Owner, Scrum Master, and Agile Development Team work together to determine what will be accomplished during the Sprint so that the team makes progress toward delivering value to the user. Sprint Planning focuses on identifying the prioritized User Stories and understanding the tasks necessary for a fixed period (e.g., 2-4 weeks for a single Sprint). The duration and scope of a Sprint must remain fixed while the Agile Development Team conducts activities.

It occurs at the beginning of each Sprint or toward the end of the previous Sprint. Sprint Plans should reflect the team’s capacity to do work, thus adhering to the Agile principle of promoting a constant development pace indefinitely. Well-defined, scoped, and prioritized User Stories are a key input to Sprint Planning. The Sprint Planning session typically takes 2–4 hours, depending on the team size and number of items that need to be worked on. The Product Owner will begin the Sprint Planning session by introducing a User Story and answering questions from team members about that User Story. Then, the Agile Development Team members will estimate and then commit to that User Story. This process will repeat until the team is at full capacity. The subset of User Stories that the team commits to deliver during a Sprint becomes the Sprint backlog. The outcome of a Sprint Planning meeting is to understand the team’s commitments for the upcoming Sprint as represented by the Sprint backlog and define the Sprint Goal.

The Sprint Goal is defined by the Agile Development Team in cooperation with the Product Owner. It reflects how the team will collaborate and focus on the highest priorities to deliver value to the user. The Sprint Goal could reflect assumptions to test, to deliver features, or mitigate risks. Questions to help define the goal include the following:

- What is the purpose of this Sprint?/What will be delivered?
- How is the Sprint goal achieved?
- Which metrics do we look at to call the Sprint goal achieved?

It is best practice for the Product Owner to review the product backlog and share the high priority items with the Agile Development Team and Scrum Master before the Sprint Planning session. Discussions during these meetings may lead to changes to a User Story (updates to the description) or identification of new stories and research spikes. It is best practice that the Sprint backlog contains all tasks (User Stories and spikes) that the Agile Development Team commits to developing during the Sprint. Agile is built on trust among the members to ask clarifying questions if there is confusion and to ensure everyone is working together to achieve the Sprint Goal/Product Vision. Changes to a committed Sprint Plan require full agreement among the Product Owner, Agile Development Team, and Scrum Master on the desired modifications while still working within the team's capacity.

Future Workforce Ecosystem Project Scenario Example

The Agile Policy Team that is developing a solution around Feature #1: *Current Organizational Description*, defines the goal for Sprint #1 as establishing a framework for capturing the current organizational structure. This Sprint Goal maps to the respective User Stories that the Agile Policy Team committed to delivering and is not just a restatement of a User Story or Feature.

4.6.2 Development

During development, the Agile Development Team focuses on designing, developing, and testing a solution, ensuring that their work achieves the stated Sprint Goal. Sprint length remains fixed throughout development. The Definition of Done, which defines what “complete” looks like for an entire increment versus a single User Story, informs what development activities must be conducted as part of the development Sprint.

The Definition of Done is defined by the team and will be used by the Product Owner to accept the developed work. Where the Acceptance Criteria ensures that the Agile Development Team has built the right product (addressed the problem), the Definition of Done ensures that the Agile Development Team built the product correctly. It is best practice to define the Definition of Done around the value that will be delivered at the end of a Sprint/Release.

After the end of the Sprint, the Agile Development Team will start a new Sprint and build upon their lessons learned and development from the previous Sprint. Continuous improvement and continuous development are key in ensuring success and help the team develop improved product outputs with every Sprint cadence. Sprint ceremonies act as forums for discussions for different topics within the Sprint, encouraging frequent and open communication of team member even outside of ceremonies. The effectiveness of the team might diminish at first as new ideas are tried out, and teammates challenge and learn about each other, but ultimately successful teams will learn to work as one.

Future Workforce Ecosystem Project Scenario Example

Members of the Agile Policy Team worked on their assigned User Stories and drafted appropriate policy language and/or a set of procedures to address the workforce's need and that fit within the constraints and Acceptance Criteria. For Sprint #1, the Agile Policy Team tested policy through peer

reviews and discussions. The team may add more substantive evaluation events, such as tabletops (desktop simulation exercises), to test the proposed policy solution in a controlled environment.

The Definition of Done incorporated the approval (sign off) from a key stakeholder, where approval represented stakeholder agreement and determination if the policy language is ready to be implemented. The Agile Policy Team defined and met the Definition of Done, shown in Table 7, to mark the completion of each User Story in Sprint #1.

Table 7. Future Workforce Ecosystem Project Definition of Done

Number	User Story and Definition of Done	
E1.F1.US1	User Story	As a Future Workforce Ecosystem Policy Writer, I want to identify the current workforce knowledge, skills, and abilities (KSAs), so that I understand the current workforce competencies
E1.F1.US2	User Story	As a Future Workforce Ecosystem Policy Writer, I want to identify the program division's organizational structure, so that I understand the interrelationships and connections among teams and departments
	Definition of Done	Review and concurrence by key stakeholders

4.6.3 Standup

A standup (or daily scrum) is a gathering of the Scrum Master and Agile Development Team. It should quickly inform the team of what is going on across the team and it is not intended to be a detailed status or technical meeting. The standup goal is that the team understands how everyone on the Agile Development Team is working together toward accomplishing the Sprint Goal and to identify impediments and risks so that the Scrum Master can address them.

The Scrum Master must facilitate the standup, and the Agile Development Team are the main participants. The Product Owner may participate in the standup, when appropriate. The standup is intended to be a short, 15-minute event. The frequency of the meeting must be determined by the team based on the Sprint length and the team's availability (e.g., daily vs twice-a-week standups).

During the standup, team members will share the following information with each other:

- What did you complete since the last standup that moves the Sprint Goal forward?
- What will you complete before the next standup that moves the Sprint Goal forward?
- What impediments do you have? [5]

Individuals will recount activities and status of work and identify items that need to be addressed outside the standup (issues that warrant discussion with the team are placed in a "parking lot"). After the standup is complete, the Scrum Master may begin to review and discuss the items in the parking lot. This discussion around parking lot items may take place with the entire Agile Development Team or with select relevant team members, including the Product Owner. The success of a standup is dependent on team cooperation, collaboration, and teamwork. If roadblocks were identified during the standup, the Scrum Master should resolve the blocker or raise the item with the Product Owner if unable to resolve.

Lessons Learned: *The frequency of standup should account for the Agile Development Team's availability and time dedicated to the development effort; a daily standup may not be feasible if team members have time split among multiple projects.*

4.6.4 Backlog Refinement

Backlog Refinement is an event that occurs continuously during development. This Scrum event serves to prepare for Sprint Planning as the team evaluates the backlog, and the Product Owner clarifies stories, as necessary, and identifies high priority items.

Backlog Refinement provides the opportunity for the team to identify dependencies and issues that may impact the next Sprint. This is typically a single session in each Sprint. The outcome of the Backlog Refinement meeting is to:

- Evaluate new items for the product backlog
- Confirm the priority of the most important items in the backlog
- Achieve an initial understanding of the Acceptance Criteria
- Gain consensus and an understanding of the User Story definition (problem definition) and users' expectations from the backlog
- Ensure User Stories align with the Product Vision
- Ensure that User Stories are ready for implementation (team is ready to select and commit to development during a Sprint).

The Product Owner should lead the Backlog Refinement and collect and incorporate input from the users and key influential stakeholders, as appropriate, to shape and prioritize User Stories. The Product Owner must determine the scale and/or definition of "priority." Priorities may be assessed based on value to the user (provides value to business or is an enabler) and importance to users, and should be informed by new information from users or lessons learned from work done by the Agile Development Team. Synthesizing inputs from various stakeholders requires understanding the stakeholder's perspective and role with respect to the product (e.g., influence versus interest). The Product Owner should look out for when the team and/or stakeholders begin creating a solution versus articulating the problem (What is needed? And why is it needed?). The Product Owner should ensure that User Stories that are ready for Sprint Planning are equal to the team's ability to deliver within one sprint. These User Stories in the product backlog are contained in the project's chosen tool. The Product Owner may remove User Stories that are no longer relevant, identify and create new stories based on feedback or what was previously delivered, emerging users' needs, and new lessons learned.

The Agile Development Team and the Scrum Master should participate in the Backlog Refinement sessions. The Agile Development Team will update estimates on how big/complex the User Story is, or ask clarifying questions that may cause them to refine the User Stories or split them to fit within the Sprint. The Scrum Master should ensure that all members of the Agile team participate, maintain the timebox, and ensure that these sessions occur frequently. Stakeholders and other SMEs are optional and may participate as needed. Figure 9 shows the specific steps and agenda that the Product Owner may follow when conducting a Backlog Refinement session.

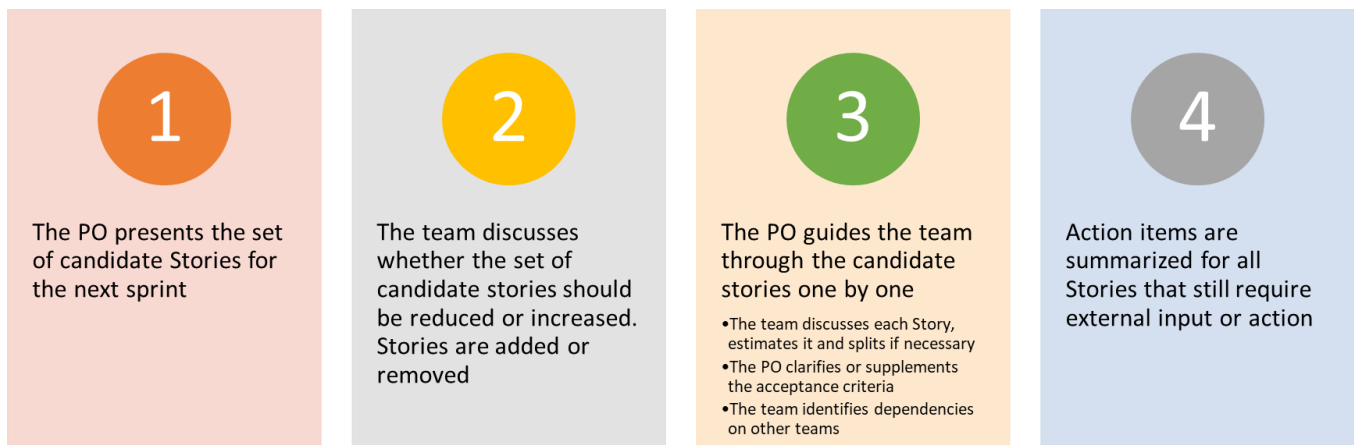


Figure 9. Backlog Refinement Session Agenda [6]

The Agile Development Team, Scrum Master, Product Owner, and relevant stakeholders and/or SMEs must work together to refine the User Stories. It is important that the Product Owner capture the discussion and update the backlog and the respective requirements, (e.g., User Story), as appropriate. For example, the Product Owner may begin addressing the following components of a User Story:

- Short title
- User Story description (As a [user] I want [goal] so that [desired benefit].)
- Acceptance Criteria
- Acceptance Tests
- Notes/Assumptions/Diagrams
- Dependencies

4.6.5 Review

A Sprint Review is a Scrum event occurring near the end of a Sprint where the Team demonstrates completed functionalities to the Product Owner, interested stakeholders, and other teammates. The Review aligns with the Agile principle that emphasizes that the highest priority is to satisfy the user's needs through early and continuous delivery of value. The expected outcome from a Sprint Review is to close out on User Stories (validate, collect feedback, and/or attain agreement from users) and to generate ideas for new User Stories for subsequent Sprints based on stakeholder feedback.

The Product Owner leads the Sprint review, and the Agile Development Team and Scrum Master demonstrate their developed products. The Product Owner may determine when to include relevant key stakeholders in the Sprint Reviews. These events are typically 60-90 minutes depending on the team size and number of items to be demonstrated. A Sprint Review should cover the following topics:

- Sprint Review and Demo Purpose
- Product Vision
- Sprint Goals and status of stories
- Review/Demo each completed User Story – emphasize value to user
- Risks (Resolved/Mitigated/Identified)
- Sprint Planning backlog for next Sprint.

The following steps serve as guidance for conducting Sprint Reviews:

- Before the Sprint Review, the Product Owner should review User Stories as they are being closed out during the Sprint (the Definition of Done and/or Acceptance Criteria has been completed per Sprint/User Story)
- Product Owner starts Sprint Review and welcomes stakeholders
- Product Owner presents meeting agenda and gives an overview (e.g., product vision, Sprint goal) and summarizes work done by teammates this Sprint. Discussions can include work that:
 1. has been done,
 2. has not been done,
 3. has been added, or
 4. removed from the Sprint.
- Agile Development Team demonstrates the work they have accomplished in the past Sprint, and answer questions—these can also be voiceovers if there is nothing to demo
- Stakeholders and Agile Development Team give feedback, which may turn into User Stories for the subsequent Sprint
- Product Owner presents the product backlog to get feedback on work for the upcoming Sprint
- Meeting ends: Product Owner can meet individually with teammates or stakeholders to deep dive into items, if necessary

The best practices for conducting Sprint Reviews include the following:

- Clearly communicate the “why” (value) of a developed product
- Identify what is most important to share and demonstrate based on who is present or will be attending the Sprint Review; make the most of opportunities to create shared understanding among the Agile Development Team and users and stakeholders responsible for approving the product, like policy
- Continuously refine what needs to be demonstrated based on which stakeholders are participating (For example, policy demonstrations may consider presenting the User Story and the User Story’s developer should demonstrate how they reached the User Story’s Acceptance Criteria)
- Continuously listen and capture feedback on the product and opportunities for improvement
- Ensure the Agile Development Team demonstrates the value of the Definition of Done.

Future Workforce Ecosystem Project Scenario Example

As part of the Sprint Review, the Agile Policy Team generated and presented the completed User Stories from Sprint #1. Future Workforce Ecosystem Project’s users and stakeholder review and provide feedback on the User Story presentations during the Sprint Review.

Specifically, for User Story E1.F1.US1 - *As a Future Workforce Ecosystem Policy Writer, I want to identify the current workforce knowledge, skills, and abilities (KSAs), so that I understand the current workforce competencies*, the team generated the following products to present at the Sprint Review:

- PowerPoint presentation with KSAs by Job Family and Band/Step Definition for the top twenty employee positions
- Spreadsheet with a list of the most prevalent KSAs found across the company
- Simple, database front end displaying KSAs Job Profile.

For User Story E1.F1.US2 - *As a Future Workforce Ecosystem Policy Writer, I want to identify the program division's organizational structure, so that I understand the interrelationships and connections among teams and departments*, the team will generate the following products to present at the Sprint Review:

- PowerPoint presentation with organizational diagrams
- Spreadsheet with a list of divisions, their interrelationships, and their connections with other divisions
- Simple, database front end displaying divisions with the most connections based on current work programs.

4.6.6 Retrospective

A Retrospective occurs at the end of a Sprint for the Agile Development Team to discuss what went well, what did not go well, and ideas for improvement for the next Sprint. The goal of a retrospective is to gain consensus on improvement ideas for the next Sprint, such as schedule changes, process improvements, understanding team capacity, work scope clarifications, and opportunities for stakeholder engagement. This event aligns with the Agile principle around regularly reflecting on how the team may become more effective and then adjusting behaviors as appropriate (a continuous improvement mindset).

The Scrum Master leads this 30-60 minute event, and the Agile Development Team must participate to provide their feedback. The Product Owner is not required to attend retrospectives but may attend as appropriate.

The following steps serve as guidance for conducting Sprint Retrospectives:

4. Scrum Master reviews—with the Team—items from the last retrospective to discuss if new ideas were successfully implemented or not, and why.
5. Scrum Master and team members individually propose “new ideas” (start), “what didn’t go well” (stop), and “what went well” (continue).
6. The Team comes to a consensus on items for the three categories with a focus on actions for the next Sprint that can also be captured in the Sprint backlog.

There are tools available to facilitate a retrospective and they include Mural (or any online whiteboarding capability for remote teams), goReflect, MS PowerPoint, and sticky notes and a board for in-person retrospectives.

Lesson Learned: *The retrospective is a team event. The Scrum Master must monitor if a change in the free flow of ideas occurs if the Product Owner attends. The Scrum Master is responsible for removing constraints for the team and may need to restrict or modify the Product Owner's attendance at the retrospective.*

4.7 Measures

Performance data may be collected and assessed at the product and team level, to assess the status and progress based on estimation techniques described in Section 0.

4.7.1 Product Measures

Agile promotes establishing product and team measures and metrics that evaluate how well planning, development, and delivery activities are performing in alignment with the goal of achieving expected outcomes. These metrics should be oriented around the product's quality and completeness, and the team's efficiency/flow, thus aligning with Agile principles around early, continuous, and frequent delivery of valuable software (or non-IT products in this paper), establishment of a sustainable development cadence, and highly motivated and collaborative team. Agile measures and metrics are different from traditional program/projects that often focus on measuring progress by adherence to a plan/schedule, as they help measure continuous improvement. Fundamentally, metrics should answer the following questions:

- Are we building the right product?
- Are we building the product correctly?
- Are we satisfying the user's needs and/or strategic initiatives?
- Are we supporting operations well?

Product measures look at measuring progress toward delivering business outcomes. Articulating Objectives and Key Results is one method to orient business outcomes around delivering a valuable product. The recommended product metrics include velocity, predictability, and progress for a Sprint or Release. Velocity represents the team's delivery capacity; the basis of this is a comparison of committed and completed Features during a Release and stories during a Sprint. The team and/or key stakeholders, including leadership, could use velocity to determine if they are adhering to a sustainable development pace. Predictability measures how well the team can plan and meet their Release objectives. It measures the relative planned versus actual value delivered. Inconsistent predictability trends may be signs of underlying issues related to resource allocations or planning. These negative trends may result in inconsistent or unreliable delivery of valuable products to the user. Identifying product measures should be part of the overarching change management or communication plan, as managers and leadership must understand the value of qualitative metrics to assess how the product meets business value.

4.7.2 Team Measures

Metrics for the Agile Development Team focus on the team's effectiveness toward incrementally delivering a value. These measures are unique to an Agile Development Team and should not be used as a means for comparing multiple team's efficiency. The recommended team metrics include velocity, burndown chart, predictability based on workflow, and quality [1], [18].

- Velocity at the team level is the basis for forecasting the team's ability to complete the planned work. Velocity is a sum of estimates for completed User Stories against committed stories during a Sprint. Increases in velocity may indicate effective teaming and collaboration, whereas decreases in velocity may indicate inefficiencies in the team's process. As the team matures and iterates through multiple sprints, previous sprint velocities can help estimate capacity in future sprints.

- Predictability is a measurement of workflow/process, looking for consistent and reliable delivery of the output. It considers the total number of items being worked on, the amount of time it takes for work to complete, and how much work is completed over a unit of time.
- Quality metrics are related to the output of the Agile Development Team, where the team may measure the number of deficiencies, the number of User Stories that pass the first time or are returned to the team for not meeting user's expectations, and the number of blockers. For a policy product example, this may include the nature and number of comments received against a section.

There are tools to support measuring metrics at the product and team level. It is a recommended best practice for the team and stakeholders to identify the measures that are most appropriate for the product and team so that everyone feels empowered to achieve the business outcomes and understands how to improve (learning mindset).

5 Operationalizing Agile Development

Operationalizing an Agile development process requires developing a roadmap that considers introducing teams to Agile, training, pilot test, and scaling. Table 8 proposes a potential pathway, which introduces and builds knowledge and skills in Agile, to practicing and applying Agile methods, to operationalizing, for organizations that are new to Agile. For organizations that are learning and applying Agile for the first time, it is best practice to train all individuals who will be involved in product development, to identify an Agile Coach, and include individuals who have previously worked on Agile teams. Agile introduces changes in behavior and organizational processes, so individuals must understand the value of developing non-IT products in an iterative manner. An Agile Coach helps organizations, teams, and individuals adopt Agile practices and methods while embedding Agile values and mindsets, quickly building credibility and trust within the team. The goal of an Agile Coach is to foster more effective, transparent, and cohesive teams, and to enable better outcomes, solutions, and products/services for users. Experience and coaching are quick ways to build rapport and credibility quickly when applying theory to actual practice.

Table 8. Agile Transition Roadmap

Major Phase	Desired Outcome	Potential Activities
Training on Agile and Scrum	To gain experience and plan for applying an Agile development process toward a specific effort.	<ol style="list-style-type: none"> 1. Identify Agile development process Leadership Champion(s) and Change Leaders 2. Provide training for Champion and Change Leaders 3. Select pilot project(s) <ol style="list-style-type: none"> a. Best practice: conduct multiple, small scale pilot testing (Consider having a variety in the types of non-IT projects to properly assess and get adequate feedback) 4. Provide training on Agile and Scrum to the managers and leadership 5. Establish a process, tools, and measures for evaluating an Agile development process 6. Develop a communication plan to socialize an Agile development process 7. Socialize an Agile development process to the broader workforce 8. Define implementation plan and prepare for pilot(s)
Build and Refine an Agile development process	To extend and mature an Agile development process and knowledge by applying the Agile development process toward a small-scale effort and a single Agile Development Team	<ol style="list-style-type: none"> 1. Build and train an Agile development team related to pilot(s) <ol style="list-style-type: none"> a. Best practice: the entire team should receive training on Agile prior to start 2. Launch pilot(s) <ol style="list-style-type: none"> a. Best practice: embed an Agile Coach for new teams b. Consider facilitating just-in-time training to dive deeper into specific Agile topics c. Best practice: pilot test on small efforts to quickly demonstrate if/how Agile could be used d. Apply an Agile development process 3. Refine an Agile development process, tools, and measures based on feedback and lessons learned from pilot(s)

Major Phase	Desired Outcome	Potential Activities
Operationalize an Agile development process	To accelerate application of an Agile development process throughout the organization by scaling efforts and teams	<ol style="list-style-type: none"> 1. Select projects to apply an Agile process 2. Scale an Agile development process to multiple teams 3. Build and train Agile development team(s) <ol style="list-style-type: none"> a. Best practice: first begin training the Product Owner so that they may begin building the Product Vision and backlog 4. Conduct and accelerate application of an Agile development process to develop a non-IT product

6 Conclusion

This paper presented a tailored Scrum process to develop nontechnical solutions and demonstrated how a fictitious team may use this process to capture and update workforce procedures in updated organizational policy. Agile is a proven method that emphasizes the value of delivering a working, usable product for users, while focusing on items of highest priority and eliminating activities that do not help achieve the desired output.

Organizations who are new to Agile and interested in applying these methods require thoughtful upfront planning to tailor known methodologies, like Scrum, to their organizational environment and to pilot methods to refine the specific adopted Agile practices. This paper presented a process and execution process around leveraging Agile for developing policy.

Regardless of domain and application, it is important to focus on following Agile principles and practices to ensure success:

- Begin with an iterative development and a small, collaborative team
- Establish trust among the team members early
- Identify Agile leaders who exemplify servant leadership – put the team first
- Continuously emphasize delivery of value – user/human centered
- Maintain a continuous improvement mindset – fail fast and often.

7 References

- [1] The MITRE Corporation, "Agile Program Management Practices for the Federal Aviation Administration Version 1.0," October 2016. [Online]. Available: <https://fast.faa.gov/docs/FAA AgileProgramManagement.pdf>. [Accessed September 2021].
- [2] J. Sutherland, "The SCRUM Fieldbook," Random House Business, London, 2020.
- [3] Beck, K. et al., "Manifesto for Agile Software Development," 2001. [Online]. Available: <https://agilemanifesto.org/>. [Accessed December 2021].
- [4] Digital.ai, "15th State of Agile Report," Digital.ai, Boston, 2021.
- [5] K. Schwaber and J. Sutherland, "The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game," November 2020. [Online]. Available: <https://www.scrum.org/resources/scrum-guide>. [Accessed December 2021].
- [6] Webber, M., et al., "FAA - Intro to Agile Training, May 17-18, 2021," The MITRE Corporation, McLean, VA, 2021.
- [7] Centers for Disease Control and Prevention, "Definition of Policy," U.S. Department of Health & Human Services, 29 May 2015. [Online]. Available: <https://www.cdc.gov/policy/analysis/process/definition.html>. [Accessed January 2022].
- [8] Birkland, T. A., An Introduction to the Policy Process: Theories, Concepts, and Models of Public Policy Making (5 ed.), Milton Park, UK: Routledge, 2019.
- [9] M. Howlett, Designing public policies: principles and instruments, Milton Park, UK: Routledge, 2011.
- [10] HM Treasury, "The Green Book: Central Government Guidance on Appraisal and Evaluation," 2020. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/938046/The_Green_Book_2020.pdf. [Accessed December 2021].
- [11] P. Sabatier and D. Mazmanian, "The implementation of public policy: a framework of analysis," *Policy Studies Journal*, vol. 8, no. 4, pp. 538-560, 1980.
- [12] Brownson, R.C., et al., "Understanding evidence-based public health policy," *American Journal of Public Health*, vol. 99, no. 9, pp. 1576-1583, 2009.
- [13] W. McKinley and A. Scherer, "Some Unanticipated Consequences of Organizational Restructuring," *Academy of Management Review*, vol. 25, no. 10.2307/259202, pp. 735-752, 3 March 2007.
- [14] The MITRE Corporation, "Innovation Toolkit: Stakeholder Identification Canvas," June 2021. [Online]. Available: <https://itk.mitre.org/toolkit-tools/stakeholder-identification-canvas/>. [Accessed December 2021].
- [15] The MITRE Corporation, "Innovation Toolkit: Stakeholder Map and Matrix," June 2021. [Online]. Available: <https://itk.mitre.org/toolkit-tools/stakeholder-map-and-matrix/>. [Accessed December 2021].
- [16] U.S. Digital Service, "TechFAR Handbook for Procuring Digital Services Using Agile Processes," U.S. Digital Service, 7 August 2014. [Online]. Available:

https://playbook.cio.gov/assets/TechFAR%20Handbook_2014-08-07.pdf. [Accessed August 2016].

- [17] Merriam-Webster, "Architecture Definition," Merriam-Webster, Inc., [Online]. Available: <https://www.merriam-webster.com/dictionary/architecture>. [Accessed January 2022].
- [18] Under Secretary of Defense for Acquisition and Sustainment, "Agile Metrics Guide Strategy Considerations and Sample Metrics for Agile Development Solutions Version 1.1," Department of Defense, Washington, D.C., September 2019.

NOTICE

This is the copyright work of The MITRE Corporation, and was produced for the U. S. Government under Contract Number 693KA8-22-C-00001, 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, Contracts Management Office, 7515 Colshire Drive, McLean, VA 22102-7539, (703) 983-6000.

© 2022 The MITRE Corporation. All Rights Reserved.