Software Acquisition Strategy: Agile Guidance

Office of the Under Secretary of Defense
For Acquisition and Sustainment

V1.2
December 15, 2019
This page intentionally left blank.
Executive Summary

Purpose

This document captures key information that should be contained in an Acquisition Strategy document for software-intensive programs that intend to use an Agile approach for acquiring software. The goal of this guidance is to be as specific and actionable as possible, recognizing the challenges to applying Agile practices within the context of Department of Defense (DoD) acquisition, and suggesting areas that should be considered to maximize success. This guidance highlights the parts of an acquisition strategy that would be most affected by Agile approaches, and while it does not aim to describe a complete acquisition strategy, elements of this guidance could be considered as part of a template for an acquisition strategy.

The goal of this guidance is to assist in developing a robust Agile strategy aimed at delivering needed capability more quickly to warfighters and other users of our systems. As such, the main difference of this guidance compared to prior versions is the focus on how the strategy will support frequent delivery of capabilities that provide value to the user / warfighter. The following sections are intended to help the program articulate:

- The PMO’s strategy for supporting continuous integration / continuous delivery of capability, in terms of addressing issues related to people (technical expertise), processes (Agile methodologies and engineering approaches), and tools (platform / infrastructure selection).
- The PMO’s intended interactions with other stakeholders in the Agile ecosystem (e.g. for requirements, Test & Evaluation (T&E), accreditation) in order to “shift left” these activities and integrate them continuously into the lifecycle.
- The contracting strategies to effectively achieve and incentivize all of the above.

Scope

This guidance is currently focused on:

- Software-intensive, Information Technology (IT) programs. Extending to other types of systems, especially embedded software systems, will be the focus of future work.
- Systems subject to DODI 5000.02. The use of Section 804 (rapid prototyping) or other non-traditional acquisition pathway is not required. As a result, the acquisition strategy may need to integrate appropriately with downstream program documentation, such as the Software Development Plan, System Engineering Plan, and Test and Evaluation Master Plan.
- Describing the strategy for a time period not to exceed five years. We recognize that software will typically continue to evolve and never reach an end state. However, we expect that planning cycles longer than the next five years are likely to involve too much guesswork, and expect that acquisition strategies should be evolved and re-certified periodically based on better information.
Assumptions

An Agile Acquisition Strategy should:

- Support a view of software acquisition as continuous engineering, in which the software never reaches a stable end state but instead is required to accommodate continual fixes, upgrades, and new capabilities.
- Rely on iterative cycles and continuous integration / continuous delivery (CI/CD) to refine over time the conception and implementation of the needed capabilities.
- Avoid heavyweight documentation, which provides a false sense of feasibility, in favor of constant monitoring and fast, evidence-based feedback.
- To the extent possible, streamline and “shift left” accreditation and T&E activities, and engage as needed with important stakeholders upfront and throughout the acquisition process.

Important note: This document describes information that should be part of an Acquisition Strategy for Agile programs. It is not intended to describe a complete acquisition strategy document. Information not called out in this document may still be required for statutory or regulatory reasons. The Services will make decisions about other elements of an acquisition strategy that are required.

As understanding of successful strategies for Agile software acquisition emerge within the DoD environment, this guidance is expected to evolve and change.
# Table of Contents

1 Purpose 6

2 Capability Need .......................... 6
   2.1 Value Statement .......................... 6
   2.2 Expected Operational Mission of this Program .......................... 6
   2.3 Threat Assessment .......................... 7
   2.4 Operational Concept Illustration .......................... 7

3 Acquisition Approach .......................... 7
   3.1 Development of Agile Competencies .......................... 8
   3.2 Agile Framework .......................... 8
   3.3 User Engagement Strategy .......................... 10

4 Technical Decisions .......................... 10
   4.1 Platform / Infrastructure .......................... 10
   4.2 Design Considerations .......................... 11

5 Program Roadmap .......................... 11
   5.1 Minimal Viable Product and Deployment Cadence .......................... 11
   5.2 Product Roadmap and Graphic .......................... 12
   5.3 Requirements Management .......................... 12
   5.4 Schedule Interdependencies .......................... 12

6 Risk Management .......................... 13

7 Metrics 13

8 Sustainment Strategy .......................... 14

9 Contracting .......................... 14
   9.1 Competition Strategy .......................... 14
   9.2 Contracting Strategy .......................... 15
   9.3 Intellectual Property Strategy .......................... 15

10 Cost and Funding .......................... 16

11 Program Office Staffing and Resourcing Profile .......................... 17
1 Purpose

Guidance:

- State the reason this Acquisition Strategy is being prepared or updated (e.g., new program, change in strategy, etc.).
- State the scope of the Agile acquisition (e.g., the Agile acquisition strategy may only apply to ground software and not to flight software for a program entering a modernization effort).

2 Capability Need

2.1 Value Statement

Guidance:

- Describe the value proposition of the capability to be provided (whether the business value or value to the warfighter) in a way that suggests measurable outcomes and is independent of specific, low-level requirements. Be specific regarding what types of users will experience the value propositions described, and whether the value proposition varies across different types of users.
- Indicate the known key operational and sustainment performance needs for this system.
- Highlight system characteristics driven by interoperability and/or joint integrated architectures, capability areas, and family- or system-of-systems.

What is different about this section for an Agile Software Acquisition: The value statement for an Agile software acquisition is by nature incremental, and value for early increments is likely to revolve around learning outcomes as opposed to delivery outcomes. Providing an incremental value statement assumes that some sense of a capability roadmap has already been thought about.

2.2 Expected Operational Mission of this Program

Guidance:

- Summarize the expected operational mission of this program, in terms of an initial set of features planned for the system.
- Identify the user(s).
- Indicate how the program fits into current and future integrated architectures.

What is different about this section for an Agile Software Acquisition: Ideally, the concept for the system has been framed in a way that is synergistic with an incremental delivery approach. A capability roadmap is sometimes used, focusing on the evolving complexity richness of mission scenarios that can be accommodated. From an architectural viewpoint, modularity and focus on external interfaces are aspects of Agile that can be reinforced here.
2.3 Threat Assessment

Guidance:

- Summarize the threat assessment in relation to the capabilities and/or operational concepts the system will support. Provide a reference to the applicable System Threat Assessment document, rather than repeating the details.
- Describe how dynamic the threat environment is, and the associated implications for the delivery tempo planned for the system. I.e., will the planned approach to deliveries provide the users / warfighters with new software versions in a manner that keeps pace with evolving threats?

What is different about this section for an Agile Software Acquisition: To the extent that the threat environment can be described in ways that address the incremental nature of capability acquisition, that description reinforces the Agile aspect of the strategy.

2.4 Operational Concept Illustration

Guidance:

Include an Illustration showing the major system components and how they work together to deliver end-to-end capability. Programs may elect to include the OV-1 diagram if available.

What is different about this section for an Agile Software Acquisition: The illustration should provide a roadmap of evolving scenarios rather than a single “complete” scenario illustration.

3 Acquisition Approach

Guidance:

- Provide information on the acquisition strategy in terms of the type of planned vehicle, and the strategy on contracting resources. Specifically:
  - Has the program made a determination on the use of a multiple award contract (MAC) such as a GWAC/BPA/IDIQ?
  - Is the program using a pre-existing vehicle, or will the program develop its own?
  - Does the program intend to sole-source or will the program include multiple contractors?
- Does the program intend to use challenge-based acquisition to allow competition among multiple contractors toward a winning design/solution?
- Explain whether the program has considered whether commercial items (COTS) can fulfill all or most of the business need with some tailoring. If so, what is the product and how has that impacted your contracting strategy to acquire a solution integrator?
- Explain whether the program has reached out to find innovative partners that can help bring new ideas to the government and foster innovation.
- Provide a general description of the iterative acquisition approach, especially focusing (if applicable) on how dependencies among hardware and software will be handled as the program progresses, especially when hardware is being acquired in a traditional large batch acquisition. For your description, consider:
How will the hardware needs of “early” (in comparison to large batch traditional software development) software implementation be communicated and accommodated to the larger program?

Will modeling and simulation, emulation, experimental hardware be used to enable small batch software development to progress without artificially creating large batches due to waiting for hardware?

- If the program will use different approaches for different categories of components / applications, describe that here. (For example, if the program includes multiple applications that will be running on a shared infrastructure, some built by the programs and others developed by external users.)

- Discuss how assessments of working software will be periodically used to gauge progress. The description should cover not just one increment but the overall approach and rationale across multiple increments.

- Describe the approach being used to coordinate with key stakeholders early and often throughout the iterative approach, particularly how to “shift left” T&E and accreditation activities. (Engagement with the end users of the system being acquired will be described in more detail elsewhere in this document.) Other stakeholders to include in shift left/continuous engagement include the contracting and financial staff, as well as those responsible for the oversight of the program.

*What is different about this section for an Agile Software Acquisition: This entire section has been re-worked to reflect Agile acquisition and implementation concepts. (So this “what is different” element is not included in some of the following subsections.)*

### 3.1 Development of Agile Competencies

**Guidance:**

- Discuss the current state of expertise in the program on using Agile methodologies.
- Based on that current state, and pipeline of scheduled arrivals, describe what is being done to address shortfalls.
  - Is the program engaging with commercial or government sources to provide coaching, consulting, or training? Will train-the-trainer methods be used to help develop skills more broadly in the workforce?
  - In what areas will SMEs be engaged to provide specialized expertise?

### 3.2 Agile Framework

**Guidance:**

- Has the program selected an Agile methodology or framework (e.g., Agile Scrum, Scrum of Scrums, Xtreme Programming, Scaled Agile Framework (SAFe)…)?
  - If so,
    - Are different methodologies being used by different teams? If so, how does the program ensure that the teams will be able to operate effectively within the enterprise?
    - Provide the rationale for that selection and describe how it will enable small batch, incremental realization of capability.
- Describe any prior experience with the selected methodology / framework within the Program/ Organization.
- Provide information on any customizations that are envisioned will need to be applied to the methodology to address organizational needs or constraints. For example, if SAFe is going to be used, but DevOps will not be included as an enabling strategy, explain the rationale and what impact that is expected to have on development and integration.
- If the program has not selected an Agile methodology / framework yet, what alternatives are being considered and what decision criteria are relevant?
3.3 User Engagement Strategy

Guidance:

- Describe how the program will engage end users and represent the user community throughout the acquisition process. In cases where there are different types of end users, explain which will be engaged and when.
- Describe how the development team will receive fast and frequent feedback from users, as appropriate, throughout the acquisition.
- Address the following areas in your description:
  - Role of the Product Owner or Product Manager\(^1\) in interacting with end users, if applicable for the chosen Agile methodology. Also, please address how the program intends to fill the Product Owner role.
  - Strategy to capture and deconflict user feedback in designing and implementing the software.
  - Approach for monitoring user engagement and user satisfaction with the value to be delivered.
  - Cost or staffing impacts of user involvement to the program, as applicable. (E.g., travel funds that would be used to ensure that program personnel and users could meet.)

4 Technical Decisions

4.1 Platform / Infrastructure

Guidance:

- If applicable, describe the program’s approach to network interconnectivity. Will the program be connected to an existing network, creating a new network, using a commercial network?
- Describe the program’s general approach to standing up or adopting an existing technical infrastructure for the development and deployment environment and DevSecOps tool pipeline (Software as a Service, Platform as a Service, Infrastructure as a Service). Describe how the selected approach will effectively support continuous integration and continuous delivery given the constraints of this program.
- Describe the criteria used to select a commercial or government-provided solution, or provide justification for any effort to develop the infrastructure within the program and show how the costs to keep the infrastructure up to date can be managed.
  - How does the solution align with industry capabilities that are available?
  - If a commercial solution will be used, has the program considered the potential for vendor lock-in?

\(^1\) In some Agile methodologies, Product Owner is the name given to the a key stakeholder, whose responsibility is to have a vision of what he or she wishes to build, and convey that vision to the team, e.g. by managing the product backlog. A Product Manager has been defined as “the person who owns the product road map, advocates for the product internally, and represents the customer in meetings with development.” (https://techbeacon.com/app-dev-testing/how-agile-distinguishes-between-product-managers-product-owners)
- Describe who on the program will have oversight and responsibility for the DevSecOps strategy and sustainment of the infrastructure.
- Discuss how the platform will be made available to subcontractors.
- Describe the general strategy for contracting for the infrastructure and toolset.

**What is different about this section for an Agile Software Acquisition**: Tooling to support both CI/CD and ongoing evolution includes tools for automated build, automated functional testing, quality attribute testing, and regression testing. Configuration management tools are also key to a CI/CD tool set. Specifying to the bidders all the toolsets that are expected to be delivered with the software is one of the success factors for Agile programs.

### 4.2 Design Considerations

**Guidance:**
- Describe the technical vision for the software, and the approach that will be taken to design.
- Indicate whether this will be brownfield, greenfield, or a hybrid software development.
- Discuss the process to be used for dealing with emergent design considerations.
- Discuss the major architectural challenges expected, and the strategy for working through them.

**What is different about this section for an Agile Software Acquisition**: Architecture considerations are key to enabling effective Agile software evolution. This is where concepts like Model-based Engineering, set-based design, and open systems architecture approaches need to be considered in light of the planned incremental delivery of capability. "Emergent design" describes the condition where implementation is actually the point at which binding design decisions are made. In smaller, simpler systems, this approach may have merit; in larger, complex cyber-physical systems, the architecture approach that serves Agile is typically to have an overall architectural vision with as little "fixed" architecture as feasible, with architecture evolution occurring at incremental delivery boundaries.

### 5 Program Roadmap

#### 5.1 Minimal Viable Product and Deployment Cadence

**Guidance:**
- Describe the program strategy for achieving the first deployment of operational software (Minimum Viable Product (MVP)).
- Describe the strategy for engaging required T&E and certification authorities in a timeframe that allows for the desired deployment of MVP and ongoing functionality. To what extent will automated testing and continuous Authority to Operate (ATO) be used as part of a DevSecOps solution to reduce the effort and time required for these activities?
- If known at this time:
  - Within what timeframe is the first operational version planned to be delivered, and why?
  - What are the plans for additional deployments of working software after that and how frequently will working software be delivered?
  - Will the selected environment support this delivery frequency?
If these decisions have not yet been made, what is the program’s approach to developing those decisions?

**What is different about this section for an Agile Software Acquisition:** In large batch deliveries, certification and T&E authorities can come in after the beginning of the development; if they are unaccustomed to working in small batches, they need to become aware of how their certifications/reports affect desired delivery cadence.

### 5.2 Product Roadmap and Graphic

**Guidance:**

- Provide the notional Product Roadmap for the current funding cycle, five years or less. Include a graphical representation along with justification. Specific considerations and/or areas of focus include:
  - Can the roadmap be used to track progress of value being delivered, understanding that the roadmap will probably change over time? (e.g., Alignment to capabilities/features being delivered over time)
  - Are there dependencies among the components/capabilities to be acquired, and are these reflected as constraints in the roadmap?

### 5.3 Requirements Management

**Guidance:**

- Describe the program’s approach to managing requirements in an Agile manner, with particular attention to the relationship between the roadmap and requirements development.
  - What level of approval (in Agile terms: Epic, Capability, Feature, Product Backlog Item/Story) will the government have for requirements and at what points in the delivery cycle?
  - How will detailed requirements be allowed to emerge appropriately?
- If the program has legacy requirements, describe how the program will avoid the risk of being non-Agile by being tied to pre-defined requirements that may not adequately reflect changing user needs or an evolving understanding of the system.
- Include any existing decomposition of requirements into Epics, Capabilities, etc., if appropriate, as an appendix to help the reader understand a snapshot in time of requirements planning.
- Describe how the work to be done will be managed and prioritized by factors such as user need, risk, cost, time required.

### 5.4 Schedule Interdependencies

**Guidance:**

- Specify the technical and/or organizational interdependencies between this program and other programs. (That is, which systems, whether existing or in development, will this software need to interface with, in order to accomplish its mission?) Discuss the relationship of the interdependencies with program activity on the critical path.
• If any memorandums of agreement are required to formalize these relationships/interfaces, describe them here. For example, identify the interface (i.e., the system this product interfaces with); the agency that owns the other system; the authority (e.g., Program Executive Officer (PEO), Component Acquisition Executive (CAE), delegated Program Manager) responsible for controlling the interface (i.e., the individual who can set the requirement; direct the solution to the interface issue; and direct who provides the funding for the solution); the required by date; and the impact if not completed.

• Describe how the program roadmap aligns with expected deliveries of capability from those programs.

**What is different about this section for an Agile Software Acquisition:** When a small-batch focused Agile program is interacting with large-batch focused traditional programs, expectations need to be set and agreed to, related to things like how incremental technical reviews and incremental documentation deliveries will be assimilated and accounted for in the large batch programs they interact with.

### 6 Risk Management

**Guidance:**

• Summarize the approach used to identify, analyze, mitigate, track, and control performance/technical/manufacturing cost, schedule, sustainment, and programmatic risk throughout the life of the program. Example risks include:
  - Obsolescence of software and hardware over the life of the program
  - Funding risks
  - Risks related to developing and deploying cross-domain solutions
  - Risks related to the need to “shift left” other key activities in order to deliver and develop capability in an Agile function, for example, risks related to whether or not a continuous ATO can be achieved.

• Describe the mechanism by which risk management aligns with the program roadmap. How are risks, once identified, potentially addressed in future iterations of the Agile process?
  - How does the governance structure allow for fast disposition of realized risk through continuous iteration / continuous delivery of capability?
  - How are risks, once identified, fed into the program backlog?

**What is different about this section for an Agile Software Acquisition:** In Agile, the primary risk that is mitigated via the Agile practices is the risk of false positive feasibility – i.e. Agile does not assume that the end point can be specified in detail at the beginning of the project, therefore the implementation data drives future design decisions vs. a priori design decisions based on assumed correctness that hasn’t been validated.

### 7 Metrics

**Guidance:**

• Describe the minimum set of high-level metrics for the program, which is planned to be provided on a regular cadence. Metrics should be able to address the frequency with which quality software is delivered into operation, user satisfaction with that software, and delivered quality, among other areas. It is recognized that metrics may need to change over the life of a program.
• Discuss how the metrics will be used to identify risks that need to be managed.
• Discuss how the measures will be shared within the team and with external stakeholders.

**What is different about this section for an Agile Software Acquisition**: Because there are more iterations of development that results in a working product, it is tempting to focus government oversight, management, and metrics on the individual teams (usually 6-10 engineers). Using the metrics from the team level as a primary measurement tactic has proven disastrous and is not recommended. Finding the “sweet spot” of measurement cadence and abstraction level is one of the challenges of Agile software acquisition.

### 8 Sustainment Strategy

**Guidance:**

• Describe how the program’s sustainment strategy, rather than treating development and sustainment as separate activities, instead treats software development and acquisition as a continuing evolution of capability, to the extent feasible within regulatory acquisition guidance.
• If the sustainment strategy involves hand-off to another organization, considerations to be addressed include:
  - How are software sustainment organizations who will be shepherding later software evolution included in key activities throughout the development, possibly holding key roles in earlier software development activities?
  - How are software environments (development, integration, test, deployment) selected and contracted for, so as to minimize drag when official transition to sustainment occurs?
  - How will software sustainment staff be trained in the tools, techniques, and environments that suppliers and the prime contractor team are using?
  - Will key enabling resources (such as a continuous ATO, if applicable, or a selected DevSecOps environment) be able to transition to the sustainment organization?

(This section should focus at the level of a sustainment strategy. The details of program sustainment planning would be included in the Life Cycle Sustainment Plan, which will be prepared and approved as a separate document.)

**What is different about this section for an Agile Software Acquisition**: Agile approaches recognize that software systems will typically continue to evolve over time, without a hard and fast separation between acquisition and sustainment. The guidance in this section attempts to resolve this by ensuring that steps taken upfront will allow the system to continue to evolve and change cost-effectively after a handoff to sustainment personnel.

### 9 Contracting

#### 9.1 Competition Strategy

**Guidance:**

• Explain how a competitive environment will be sought, promoted, and sustained continuously.
• Consider whether a modular contracting strategy may encourage competition, by focusing on discrete needs (i.e., cloud infrastructure, licensing, Agile development teams, microservice solutions, etc.) rather than a single contract to provide all needed solutions. This may include working with small companies with innovative technologies. If relevant, discuss the small business engagement strategy and how such engagements will be supported.

**What is different about this section for an Agile Software Acquisition:** Using a modular contracting approach may promote competition by letting smaller, shorter contracts. Managing a portfolio of contracts supports an evolving acquisition strategy for IT solutions. A modular approach requires the government to manage the enterprise architecture and ensure all contracts align to a common architecture.

### 9.2 Contracting Strategy

**Guidance:**

- Agile development solutions can be acquired using a variety of contract vehicles to support a program through the system lifecycle. Contract type selection (e.g., Firm-Fixed-Price; Cost; or Time and Materials) will be informed by the selected contract vehicles and allowable contract types for each (i.e., cost type contracts are not allowable when using FAR 8.4 – Federal Supply Schedules or FAR 12 – Commercial Items) and required solution (i.e., tools, licenses or development team services).

- Discuss how contracts can drive:
  - An innovative culture;
  - Outcomes and mission driven over specific requirements

- Explain how cost and value delivery progress will be measured and reported (e.g., EVM or Agile functional equivalents) and what contract deliverables will be required

- Discuss whether the program plans to use contract incentives to incentivize Agile practices on the part of the contractors.

**What is different about this section for an Agile Software Acquisition:** Although a variety of contracting strategies can be used to support Agile acquisition, they all need to support an iterative approach and have sufficient flexibility to accommodate changes to scope or size that occur due to learning from the iterations. Contract types that might not have been widely used in past for development efforts, such as time and material contracts, may provide flexibility to respond to changes expected during Agile development. Product and service contracts may both be needed as part of a comprehensive Agile acquisition strategy. Also, Agile approaches may pose distinct challenges for monitoring and evaluation contractor performance when the desired software capability is allowed to shift or evolve over time.

### 9.3 Intellectual Property Strategy

**Guidance:**

Summarize the Technical Data Rights Strategy for meeting product life-cycle data rights requirements and to support the overall competition strategy. Include:

- Strategy by which the government will have access to (not necessarily ownership of) the source code to support future analysis and use. This strategy should include mechanisms to stipulate and enforce that proprietary software not be included in the software baseline without PM office approval, and that it
be included in the baseline in a manner that preserves modularity of the software and does not unnecessarily intermix proprietary and non-proprietary software.

- How the government will have access to information sufficient to have technical insight into what code goes into the build of the software, such as: build automation software, engineering data, drawings, models, and Bills of Materials (BOM).
- Additional sources of software data to support the product support life cycle strategy, which may include baseline documentation data, analysis data, cost data, test software and harnesses, test data, results of reviews, performance data. Consider data that would be needed to support activities such as training, Information Assurance protection, open architecture, configuration management, engineering, technology refreshment, and reliability management.

When the government takes delivery of software source code, the government should ensure that all additional artifacts necessary to compile, test, secure, deploy, and operate the software are delivered as well. These artifacts can include (but may not be limited to): scripts, tools, databases, libraries, and other software executables.

**What is different about this section for an Agile Software Acquisition:** In an Agile context, the government needs to have sufficient insight so as to evaluation progress and quality in the absence of hard and fast measures against a static baseline. Also, the government needs access to sufficient information about the codebase to allow evolution of the code to happen over many iterations of the software.

## 10 Cost and Funding

**Guidance:**

- Summarize the program funding levels by year, for at least the next two years. Discuss costs for:
  - Labor: Anticipated development team needs over an initial period of time (number and size of teams, what mix of government and contractor personnel), subject to update as the program evolves
  - Equipment and infrastructure (Cloud hosting, DevSecOps and other tools, etc.)
- Discuss the costing strategy used to produce the estimates.
- Discuss whether the program is fully funded in the Future Years Defense Program (FYDP).
- Discuss whether the funding will be provided across multiple appropriations categories, and whether that aligns with the program roadmap.

**What is different about this section for an Agile Software Acquisition:** An Agile budget would be unlikely to be built up against a set of detailed requirements but instead look more like a Level of Effort plan where a certain amount of capability is required to adapt and evolve software. Also, when funding is distributed across different appropriations categories it can hamper the flexibility needed to respond to user priorities, which may require a mix of enhancements, bug fixes, vulnerability resolutions, or new capabilities.
11 Program Office Staffing and Resourcing Profile

Guidance:

- Provide a time-phased workload assessment identifying the manpower and functional competencies required for successful program execution. Elaborate on how this level of resources was estimated as sufficient for delivering capability at the needed cadence. Considering the overall, technical, acquisition, sustainment, and management approach, specify the number of personnel who are required to manage, oversee, and execute this program.

- Specify the number of teams, the size of teams, and the purpose of each team for supporting frequent delivery of software capability.

- Include a projected manning profile based upon the overall approach and program schedule for government, Systems Engineering and Technical Assistance, and Federally Funded Research and Development Center(s) support.

- Discuss what will be done to ensure that teams have the proper skillsets to support the decentralized decision-making necessary for the program to take advantage of fast learning cycles in a continuous integration / continuous deployment environment.

What is different about this section for an Agile Software Acquisition: Agile approaches typically require teams to be self-organizing, meaning that they need to have access to a mix of skillsets. In addition, the availability of expertise over time needs to support the level of effort-based cost approach described in the previous section.