DoD's Software Acquisition Pathway

Digital Delivery at the Speed of Relevance

Joint IT & SW Cost Forum

Sean Brady
DoD Senior Lead for SW Acq
USD(A&S) Acquisition Enablers

Matt MacGregor
DoD Acquisition SME, MITRE

https://aaf.dau.edu/aaf/software/

SWP Adoption Snapshot

DEFENSE

DOD procurement lead says software acquisition changes are yielding results

Stacy A. Cummings delivers a keynote speech to the Naval Postgraduate School's 18th Annual Acquisition Research Symposium, virtually, the Pentagon, Washington, D.C., May 11, 2021. (DOD photo by U.S. Air Force Staff Sgt. Brittany A. Chase)

In April, Kessel Run All Domain Operations Suite (KRADOS) was declared a **minimal viable product (MVP), in accordance with the definition outlined in the new DoDI 5000.87. "This is a huge milestone for Kessel Run, ACC and our users,"**

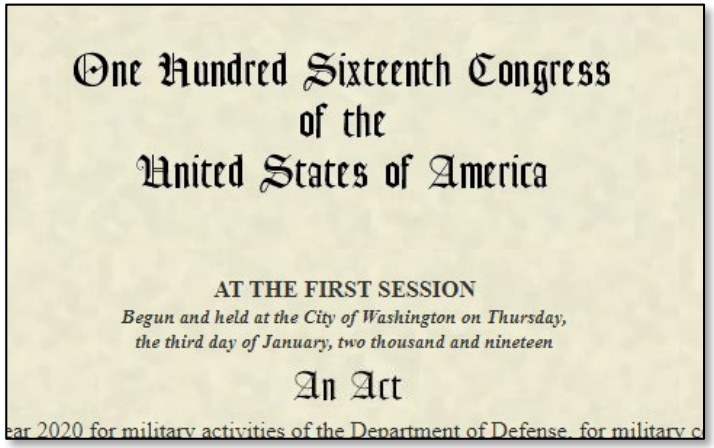
- Early in SWP journey
  - Many are in Planning phase
  - Mature programs moving direct into Execution
- Small and ultra-large scale adoption
- **Transitions**: MTA to SWP
- **Hybrid**: [MCA + SWP] & [MTA + SWP]
- **Diversity**: Business Systems, Cyber-Physical, C4ISR
- Portfolio-level adoption emerging
- PEO-level adoption

FY20 NDAA Section 800

Directed DoD to create two software acquisition pathways

Applications and

Embedded Systems



One Hundred Sixteenth Congress
of the
United States of America

AT THE FIRST SESSION

Begun and held at the City of Washington on Thursday, the third day of January, two thousand and nineteen

An Act

ear 2020 for military activities of the Department of Defense, for military c

- Software programs shall not be treated as an MDAP
- Exempt from JCIDS (unless VCJCS, A&S, SAEs agree on new process)
- Streamline SW requirements, budget, acquisition processes
- Demonstrate viability and effectiveness of capabilities for operational use within one year after funds first obligated

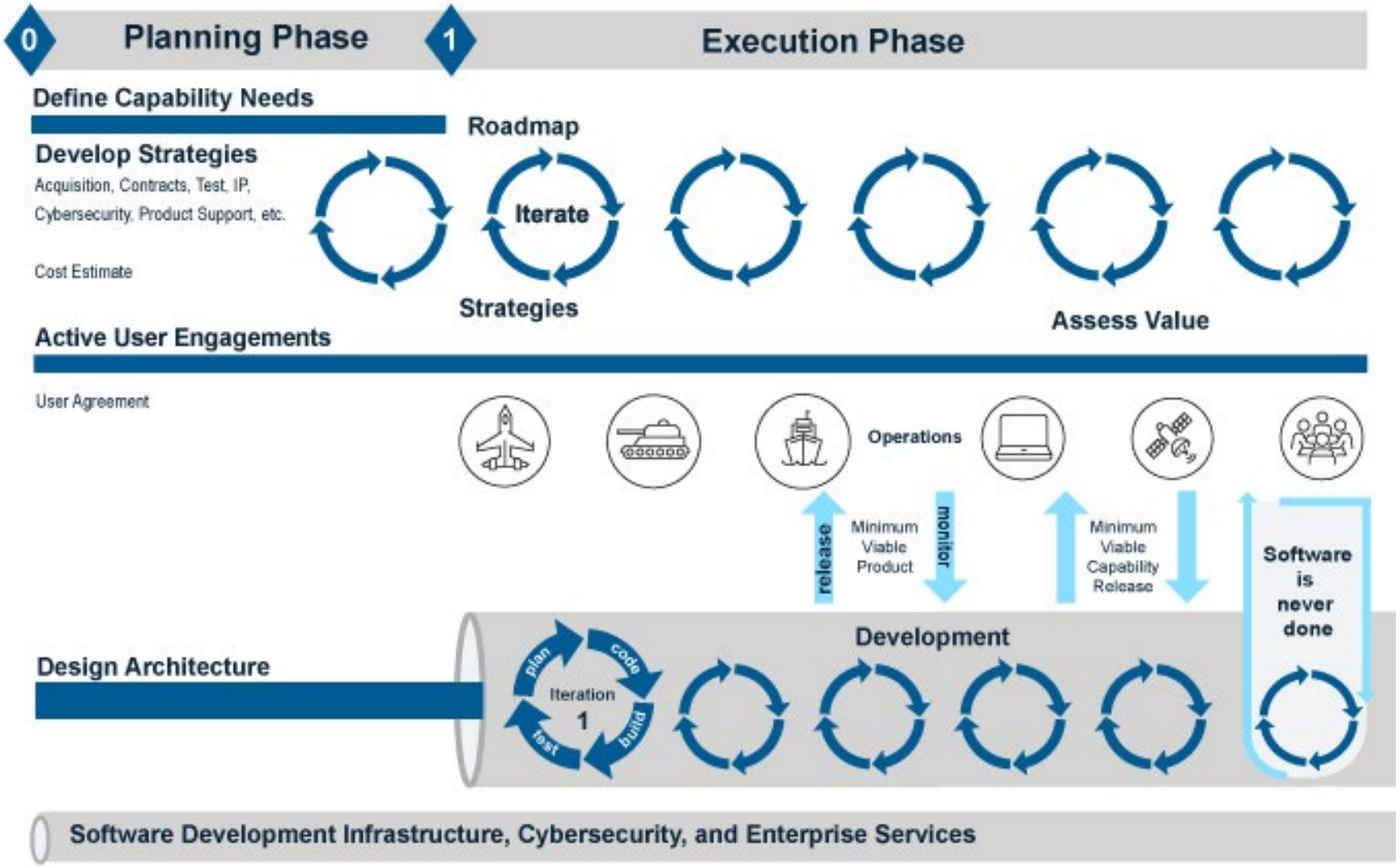
https://www.congress.gov/bill/116th-congress/senate-bill/1790/text

Key Elements of SW Acquisition Pathway

- Modern SW development practices
- Human-centered design
- Active, committed user engagement
- Enterprise services/platforms
- Rapid and iterative deliveries
- Gov't-industry software teams
- Automated tools

Source: DODI 5000.02 Section 4.2

Software Acquisition Pathway



https://aaf.dau.edu/aaf/software/

Strategy Development in SW Pathway

0 Planning Phase

Develop strategies and required artifacts for the DA to approve the program to begin Execution Phase.

Explicitly NOT a traditional acquisition milestone with dozens of major documents required.

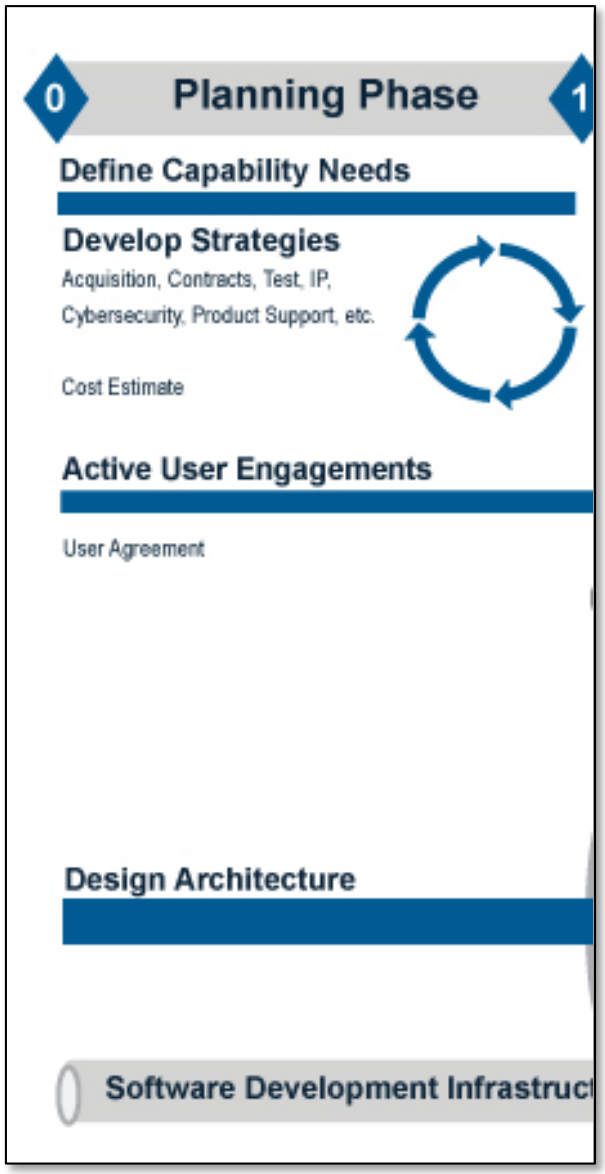
1 Execution Phase

Continuous improvement of strategies based on user feedback, team and system performance, shifting priorities, integrating best practices, etc.

Program documentation should be constrained to what is needed to effectively manage the program.

https://aaf.dau.edu/aaf/software/program-management/

Planning Phase

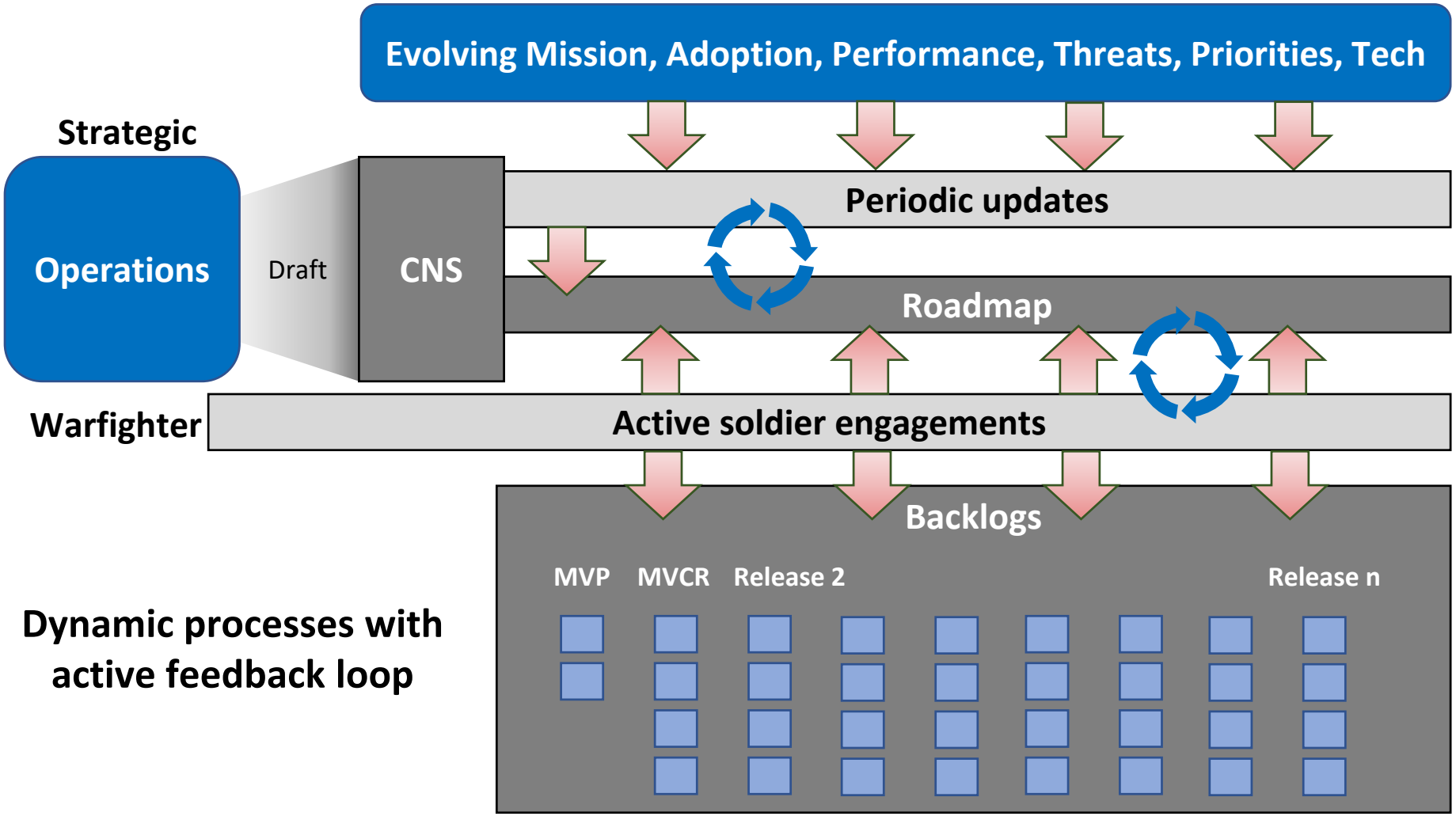


Focuses on understanding the users' and systems' needs and planning the approach to deliver capabilities to meet those needs

Key Artifacts

- Capability Needs Statement
- User Agreement
- Program Strategies
  - Acquisition Strategy
  - Contracting Strategy + IP Strategy
  - Test Strategy + Cybersecurity Strategy
  - Product Support Strategy
- Cost Estimate

https://aaf.dau.edu/aaf/software/planning-phase/



Dynamic processes with active feedback loop

Capabilities Needs Statement (CNS)

A high-level capture of need with enough information to define the software solution space and consider the threat environment.

- Sponsor and Requirements Manager ID operational software capabilities needed
- Draft CNS to start the Software Pathway
- Refine during Planning Phase and approve prior to entry into Execution Phase

A&S Acquisition Enablers shop collaborating with Components to encourage adoption of flexible and streamlined requirement processes for the SWP.

Clear Understanding of What is Needed

https://aaf.dau.edu/aaf/software/cns/

Draft CNS Template

User Agreement

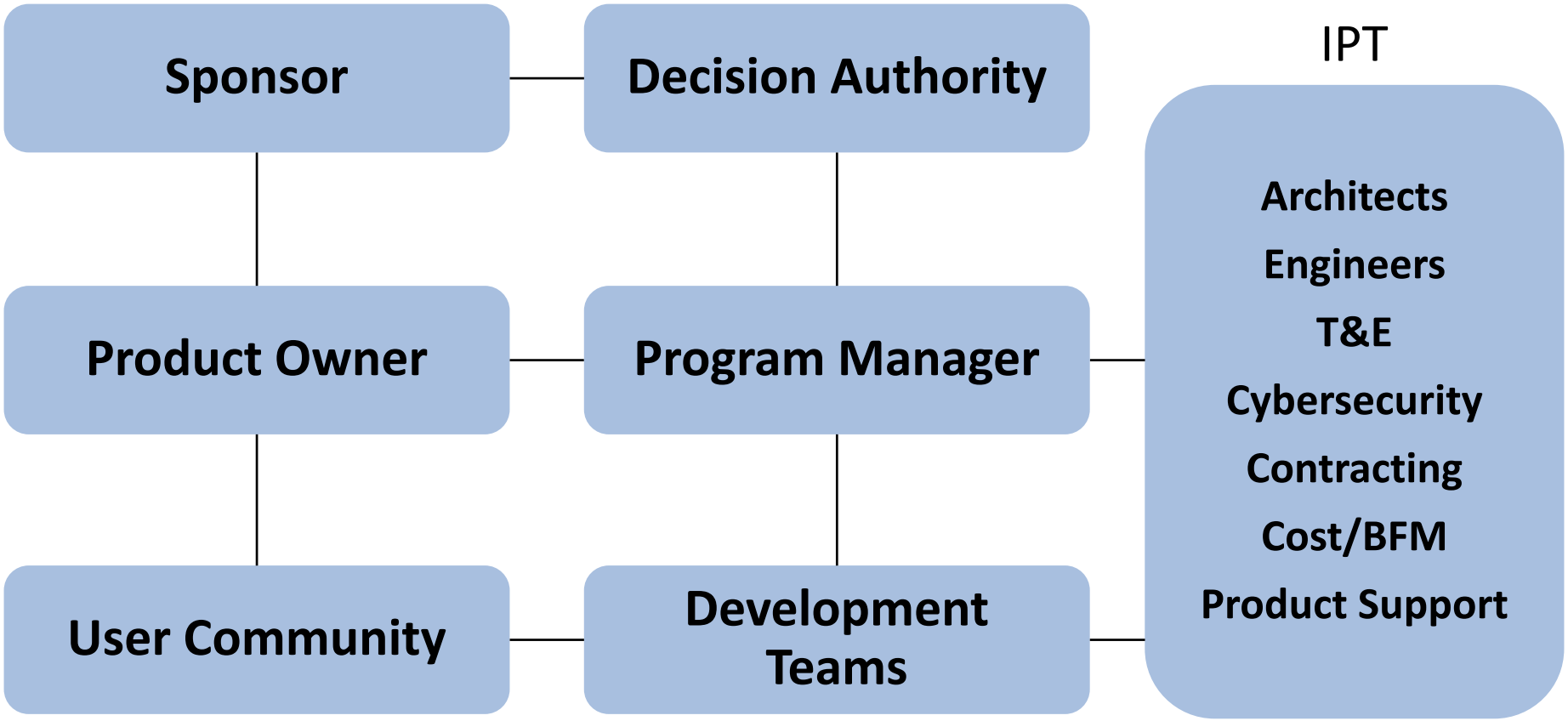
Agreement between the operational and acquisition communities to ensure active user involvement and informed decision making.

- Ensure proper resourcing of user involvement to support development
- Commit to active user involvement throughout design and development
- Clarify roles for requirements management and user feedback

Establish Strong Ties to Users from Start

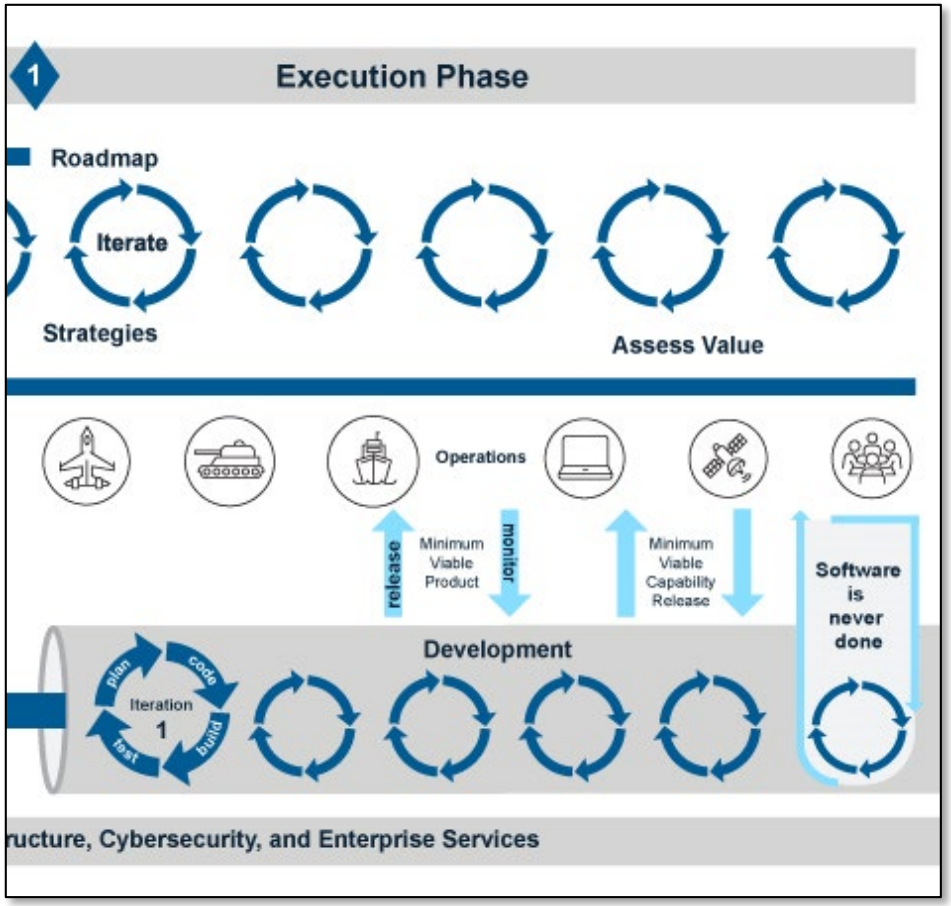
https://aaf.dau.edu/aaf/software/user-engagement/

Draft UA Template



Integrated Teams Across Operations and Acquisition; Government and Vendors; All Functions and Levels

Execution Phase – Key Activities



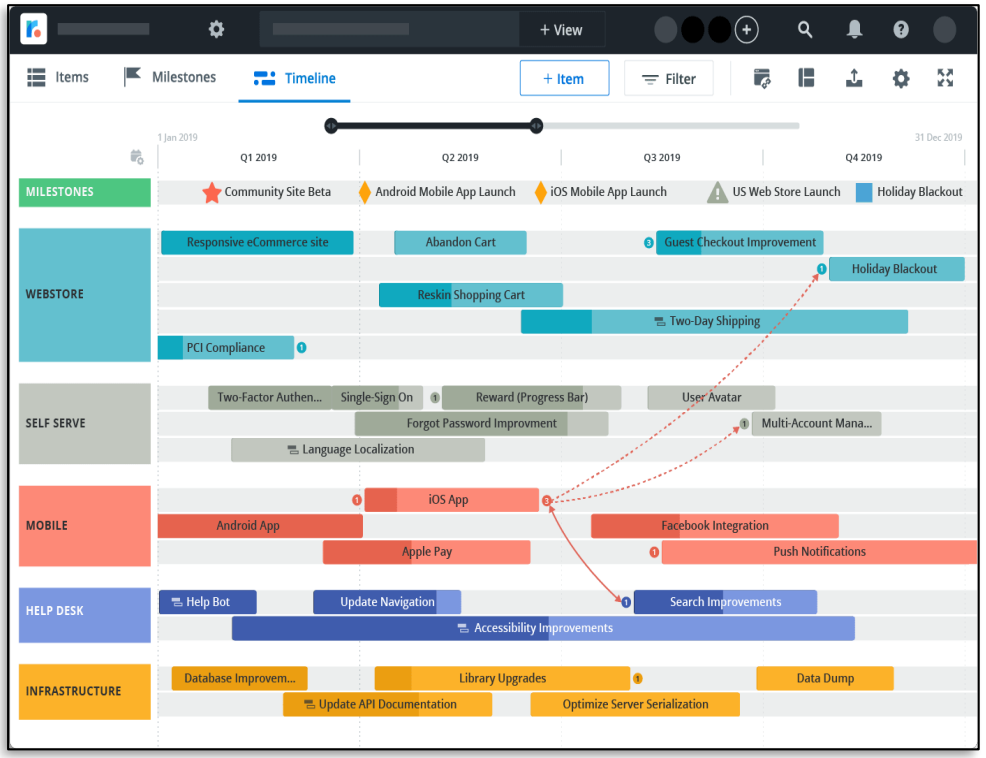
- **Product Roadmap**
- **Program Backlogs**
- **Active User Engagements**
- **Develop, Deliver Software**
- **Track Metrics**
- **Value Assessments**

Continuous improvement to maximize mission impact.

https://aaf.dau.edu/aaf/software/execution/

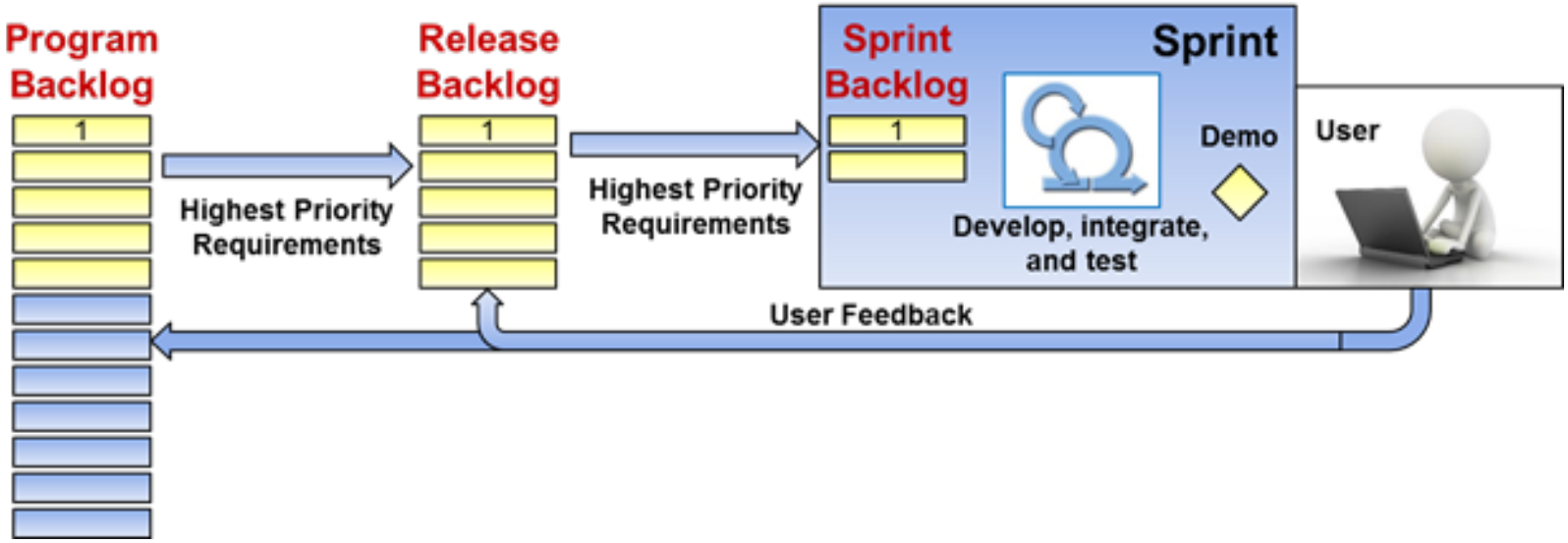
Product Roadmap

- High-level visual summary that maps out the vision and direction of product offerings over time
- Describes the goals and features of each software iteration and increment



Can use electronic tools to support roadmap creation and management to provide stakeholders clear and up-to-date status of the near-term plan

Program Backlogs



- Identify detailed user needs in prioritized lists
- Allow for dynamic reallocation of scope and priority of current and planned software releases
- Issues, errors, and defects identified during development and operations should be captured in the program's backlogs to address in future iterations and releases

Metrics and Reporting

- **Program Registration (within 60 days of joining SWP)**
  - High level info on program (meta data)
- **Insight Reporting (every 6 months Apr/Oct)**
  - Requirements basis
  - Key dates (planning, execution, FFO, MVP/MVCR)
  - Cost Estimate (high level)
  - Contract Strategy (high level elements from drop down)
  - Cyber Resilience (ATO lead time, MTR cyber event, etc.)
  - Performance Metrics (frequency, lead time, fail rate, MTTR)
- **Program Management Metrics**
  - For PMs to assess program health and progress
  - Tailored to each program – share with stakeholders
  - Maximize use of automated tools to track/report

Value Assessments (VA)

- **Rigorous form of modern management and risk mitigation**
- **Fail fast, fail cheap and dynamically and continuously inform further investments**

- In the SWP: PM has new type of "contract" with the DA and Sponsor:
  - iteratively deliver value assessed through a VA

- VA reveals how much impact the SW has on mission from end user's perspective
  - in short-term (immediate functionality)
  - long-term (designing or refactoring for future capability)
  - and given the funding provided to the SW development effort

- Subjective & objective measures measure overall value achieved for assessment cycle
  - measures identified each assessment cycle using governance process in User Agreement
  - VAs consider more than just the capabilities that are immediately visible to the user

- PM and DA look at investment and outcomes produced and either pivot or persevere.

Faster feedback loop allows build, measure, learn cycles which are inside the OODA loop of a traditional APB process

Objective Assessment:

ID Range	From 50km to 70km	80km	Exceeded Goal. Can identify targets 30km farther, increased engagement opportunities by x%
Accuracy	From 60% to 70%	80%	Exceeded Goal. 20% more reports accurate, reduced r of fratricide by x%
Operating Time	From 100 hours to 150 hours	150 hours	Met Goal. New software improves power utilization, and increases operating tim
		Value Assessed	**High Value**

Deployment Frequency	6x/yr For Highest Prioritized Features	4x/yr Mostly Highly Prioritized Features	Did Not Meet Goal. The releases delivered however provided important capabili
Change Fail Rate	<6%	10%	Did Not Meet Goal. The program still achieved reasonable fail rate levels.
		Value Assessed	**Moderate Value**

Subjective Assessment:

Usability Improvements to Critical Functions	Assessment
Synchronization of Radar Arrays	The ability to synchronize radar arrays quickly a easily upon system startup was substantially improved. Users reported that the time to conduct the synch was reduced and the synch procedures for executing the synch were easier accomplish.
Switching Between Control Modes	The ease in switching between control modes v substantially improved. Users reported that th User Interface to execute this function was mo intuitive and required fewer steps than in previous configurations.
Value Assessed	**High Value**

SW Delivery	Key Capabilities	Cost
Release 5	ID Range and Radar Array Synchronization Improvements	$16M
Release 6	Accuracy, Operating Time and Control Mode Switch Improvements	$17.5M

Funding Expended During Assessment Period: $33.5M

Program Deliveries

During the assessment period, the program office delivered Releases 5 and 6 on the Product Roadmap briefed at the last governance meeting. These releases addressed xx key priorities, added XYZ functionality, addressed 3 critical cyber vulnerabilities, and improved program performance from X to Y. **The costs for each release are captured above**

Value Assessment Rating: **High Value**

Value Assessment Narrative: program successfully developed and released SW that was timely, provided significant improvements for operational users and was **worth the investment**. The **improvements** to ID Range, Accuracy and Operating Time are **substantial** and will result in more effective military operations. The **usability** improvements to the critical functions of synchronizing radar arrays and switching between control modes were **substantial**. The **program does still need to mature** its SW dev pipeline to deliver more frequent releases with higher quality code. **Overall**, the **user community** is greatly pleased with the value received over the last assessment cycle and **recommends continuing to fund this effort at the requested levels.**

SWP Baselines and Progress

SWP programs should NOT baseline cost, schedule, and performance using traditional approaches. APB is NOT required and highly discouraged.

Legacy HW Centric Systems

- Define requirements upfront
- Detailed cost estimates
- Baseline C/S/P in APB
- Measure performance vs APB
- Track contractor via EVM data
- Focus on compliance

Modern SW Practices

- Iterative requirements
- Active user engagements
- Iterative cost estimates
- Performance via working SW
- Annual value assessments
- Continuous improvement
- Responsive to changes
- Focus on users/mission impact

DODI 5000.87 does NOT require an APB
Programs migrating to SWP should sunset their APBs

Contracting Considerations

Instead of a single monolithic contract for software solution

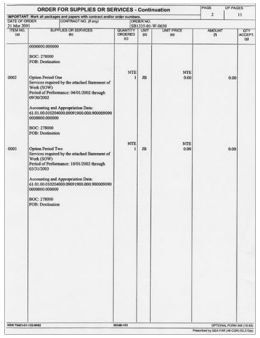
Portfolio of contracts of using *Modular Contracting

Example Modular Contracting Strategy

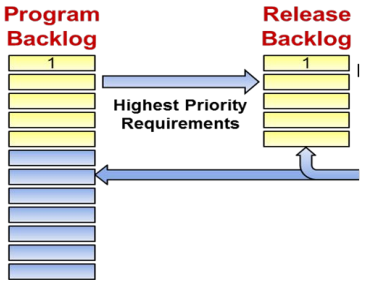
	Contract Strategies
Agile S/W Dev Team(s) (Services)	FAR 8.4, FAR 12, FAR 13.5, FAR 16.5
Microservice Solutions (Tools)	FAR 8.4, FAR 12, FAR 13.5, FAR 16.5
DevSecOps-aaS (Manage CI/CD Pipeline)	FAR 8.4, FAR 12, FAR 13.5, FAR 16.5
Platform-aaS (CI/CD Pipeline)	FAR 16.5, BOAs (i.e., Platform One)
Infrastructure-aaS (Cloud solution)	FAR 16.5 (i.e., Cloud One, AWS GovCloud)

Agile Software Dev Contracts

(may have separate contracts for each dev team)



Contract Requirements
Tasks and activities describing Agile S/W development needs to develop and deliver solution



Agile Requirements
Capabilities expressed in user stories and continuously managed in backlogs to achieve solution required by contract

*FAR 39.103

SWP and AAF Lessons Learned

- Fast moving programs view **the SWP as a natural home**
- **Faster is possible**: more SW programs using SWP practices
- **Clarity of purpose** and objectives – **tailor to achieve**
- **Empowered decision authority** and **program manager**
- Getting **broader stakeholder buy-in** on new model
- Balance **speed with rigor**
- **Demonstrate early success** – garner additional support
- **Continuous improvement** – program/enterprise levels
- Leverage common strategies, platforms, SMEs

Ignite Innovation and Execution

Partner with Services and Joint Staff to streamline and tailor requirements processes for software

Partner with Services and CAPE to streamline and iterate on software cost estimation

Partner with Services and DOT&E, DT&E to modernize, integrate, and automate software T&E

Each Ignite effort rooted in real-world: pilot programs driver of policy formation

DoD Services/Agencies Empowered and Directed to Align and Streamline Processes

What Do SW Programs Need from a Cost Estimate

1. **Timely**. Support short decision timelines; move rapidly from planning to execution; stay responsive to dynamic shifts

2. **Informative.** Provide general costs across various elements to assist ongoing planning.

3. **Defensible**. Defend budget request to resourcing agents.

Common Cost Estimate Challenges

- Addressing different types of SW programs
- Long-range planning limitations
- Blurred life cycle phases and program level ‘definition of done’
- No generally recognized standard unit of measurement
- Aligning cost estimate processes to support SW delivery cycles

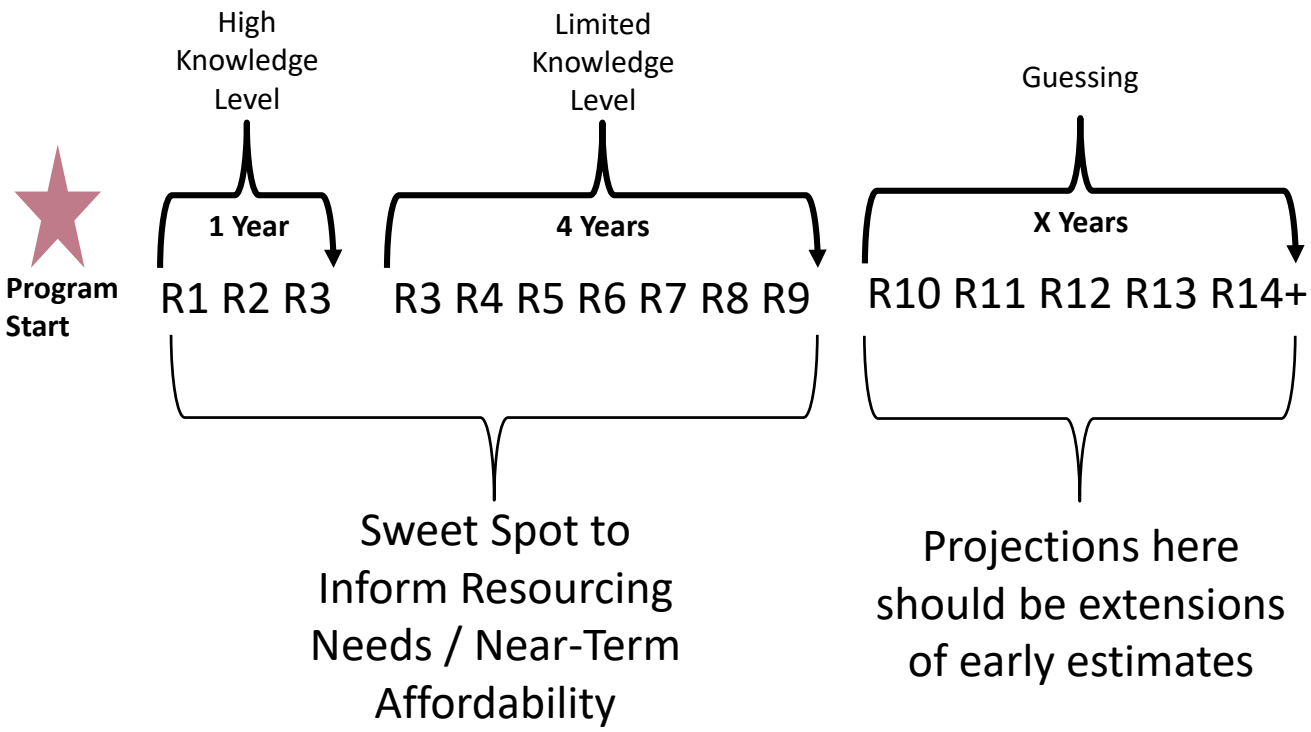
Challenge – Different Path Considerations

Application Path More Straightforward Stable Costs	Rapid development and deployment of software running on commercial hardware (including modified hardware) and cloud computing platforms.
Embedded Software Path More Complex, Dynamic Costs	Rapid development and insertion of upgrades and improvements for software embedded in weapon systems and other military-unique hardware systems.

- Not baselined
- Cloud-Based Infrastructure
- Seamless Fielding
- High Velocities

- May be partially baselined
- Mixed Infrastructure
- Fielding More Complex
- Lower Velocity (initially)

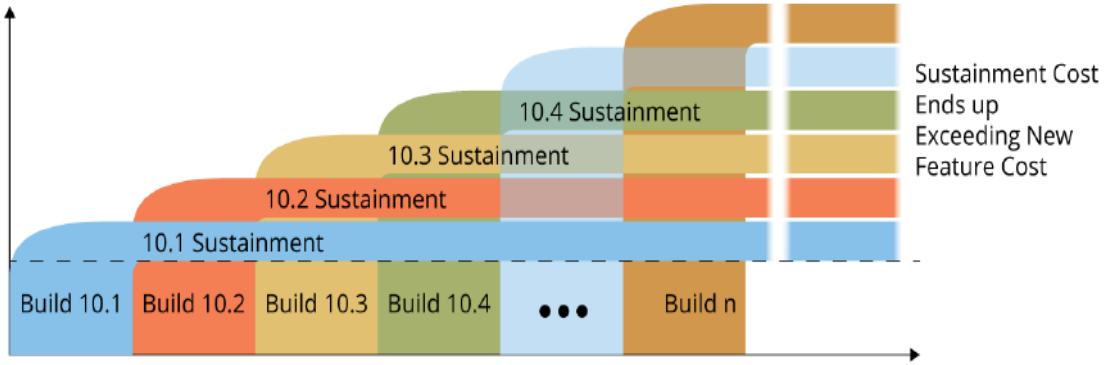
Challenge – Long-Range Planning Cycle



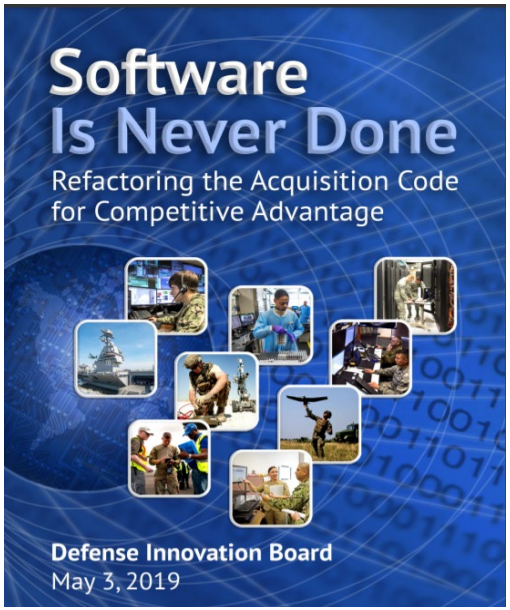
This shift is still being understood across the acquisition system that historically has been prediction-based.

Challenge – Blurred Lifecycle

- **Moving SW programs away from legacy sustainment practices – too costly**
- **Focus needs to be on establishing a continuous software pipeline until user demand signal ends.**
- **Appropriation rules may still drive use of different colors of money**



DIB SWAP- Layers of Sustainment



Challenge – Supporting Faster SW Cycles

Estimating methodology is refined over time; initial estimate based on simple models where the data is well understood

Initial Estimate	Annual Updates
Planning Phase	**Execution Phase**
Early estimation techniques with high level adjustments for Agile practices	Once development beings, a cadence is established, the estimate can be refined

Cost estimate should be developed based on the available information – should not be on the critical path to execution phase.

Software releases provide more opportunities to iteratively refine the estimate based on learning, understanding the customer needs and performance of the development teams.

PROPOSED COST ESTIMATING PROCESS USING SWP ARTIFACTS

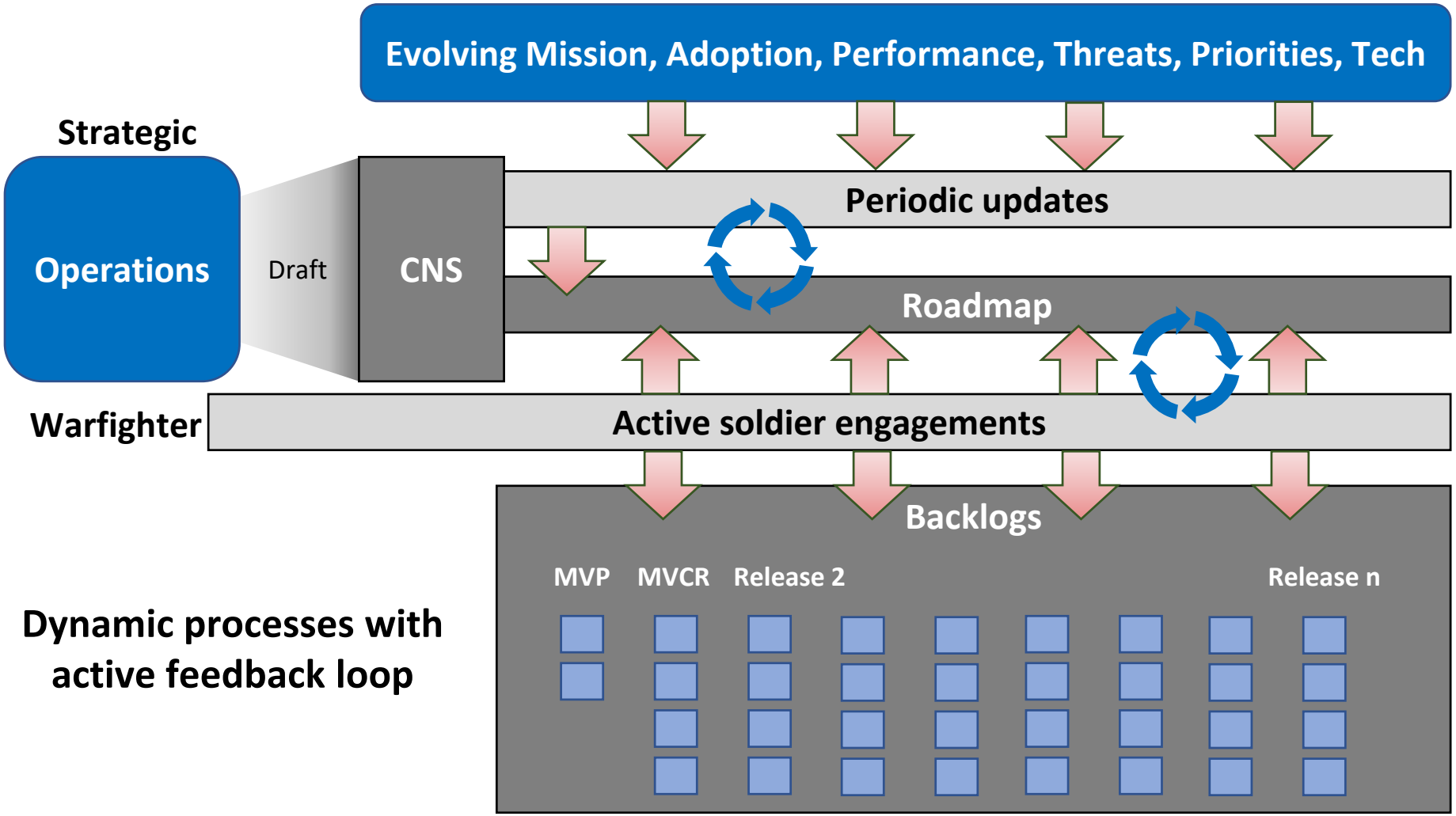
GAO Guidance

Table 11: 12-Step Cost Estimating Process and Agile Cadence Examples

12-Step estimating process step and definition	Agile environment and the GAO cost estimating process
Step 1: Define the estimate's purpose The purpose of a cost estimate is determined by its intended use, which determines its scope and detail.	During release or initial planning, determine how any cost estimates will be used.
Step 2: Develop the estimating plan This step involves selecting the estimating team members and developing a schedule that includes enough time to perform all steps commensurate with the estimate's purpose.	During initial planning, identify the cost estimating team that will develop the estimate and any technical experts that will be needed to support the estimating effort. The estimate plan should also include details about when the government program office plans to update the estimate with Agile metrics.
Step 3: Define the program Program personnel identify the technical and programmatic parameters on which the team will base the estimate. This information should be kept updated at all times so that it remains current.	These steps (steps 3-7) should first occur during initial program planning with the development of a road map or vision and be updated as the estimate is refined at established intervals, such as after a release, in support of program milestone reviews, or whenever there are updates to the road map. Agile performance measures and artifacts such as burn up/burn down charts, velocity metrics, and the product backlog can be used to update the estimate accordingly. It is important that the cost estimating team is integrated into release planning so that team members can fully understand the changes to the plan and update the estimate to reflect those changes that occur naturally during the Agile process (e.g., additional detail is provided through a requirements decomposition process). An independent cost estimate should be developed after the initial cost estimate and then repeated at major milestone reviews for the program. Between estimates, a sufficiency review of the cost estimate after major updates should be performed to assess the credibility of the program office estimate.
Step 4: Determine the estimating structure This step defines at various levels of detail what the program needs to accomplish to meet its objectives. Typically, estimators will have access to a work breakdown structure (WBS) that decomposes the work into a product-oriented, hierarchical framework supplemented by common elements like program management, systems engineering, and systems test and evaluation, etc. A WBS promotes accountability by clearly identifying work products and enables managers to track technical accomplishments. It also outlines how program elements progressively subdivide into more detail as new information becomes available.	
Step 5: Identify ground rules and assumptions The estimating team establishes ground rules that represent a common set of agreed to estimating standards such as what base year the team will use to express costs, the number of expected program quantities, and the anticipated contracting strategy. When information is unknown, the estimating team must fill the gaps by making assumptions so that the estimate can proceed. Because many assumptions profoundly influence cost, management should fully understand the conditions the estimate was structured on. Well-supported assumptions include documentation of their sources along with a discussion of any weaknesses or risks.	
Step 6: Obtain the data The team collects, normalizes, documents, and archives the cost, schedule, programmatic and technical data it will use for the cost estimate.	

Key Takeaways

- Use Roadmaps
- Refine Estimate Over Time
- Update using Metrics and Backlog
- Integrate Cost Team



Dynamic processes with active feedback loop

Start with the Capability Needs Statement

Capability Area

High-level groupings of enduring needs which will be met over a series of software releases.

Capability Requirement
Ability to Plan Collaboratively with Key Operational Stakeholders
Synchronize Execution Across All Domains
Monitor Execution, Assess Effects, and Adapt Operations
Communicate Commander's Intent and Guidance
Establish/Adapt Command Structures and Enable Collaboration
Exercise Command Leadership
Leverage Mission Partners

Performance Attributes

Includes Performance Measures and Target States. Should articulate what is valued by the operational community to satisfy gaps in the capability areas.

Backlog Decomposition

The team should decompose the capabilities articulated in the CNS into an **initial** backlog of unprioritized items (stories, features) that will need to be addressed.

At the planning phase of a new effort, this may be the entirety of the information available to a cost estimator – only execution will provide greater insight

Craft Initial Estimate (During Planning Phase)

- Given the immature state of the backlog prioritization and resulting roadmap, the estimating team will need to be tightly coupled with the government development team to develop an initial estimate. May be:
  - **Driven** by available analogous efforts (if available)
  - **Driven** by likely available near-term resources (plan X number of dev teams)
  - **Driven** by government team's ability to manage X dev teams
  - **Driven** by expected content of the MVP (which will inform future refinements)
  - **Driven** by operational sponsor's expectations/urgency for the program
  - **Driven** by initial fixed stand-up costs: CTR O/H, Gov facilities, development infrastructure, SW licenses, test tools/configuration, training, coaching labor.

Expect Initial Estimate to be Capacity-Focused as Users Become Engaged and Prioritize Future Work

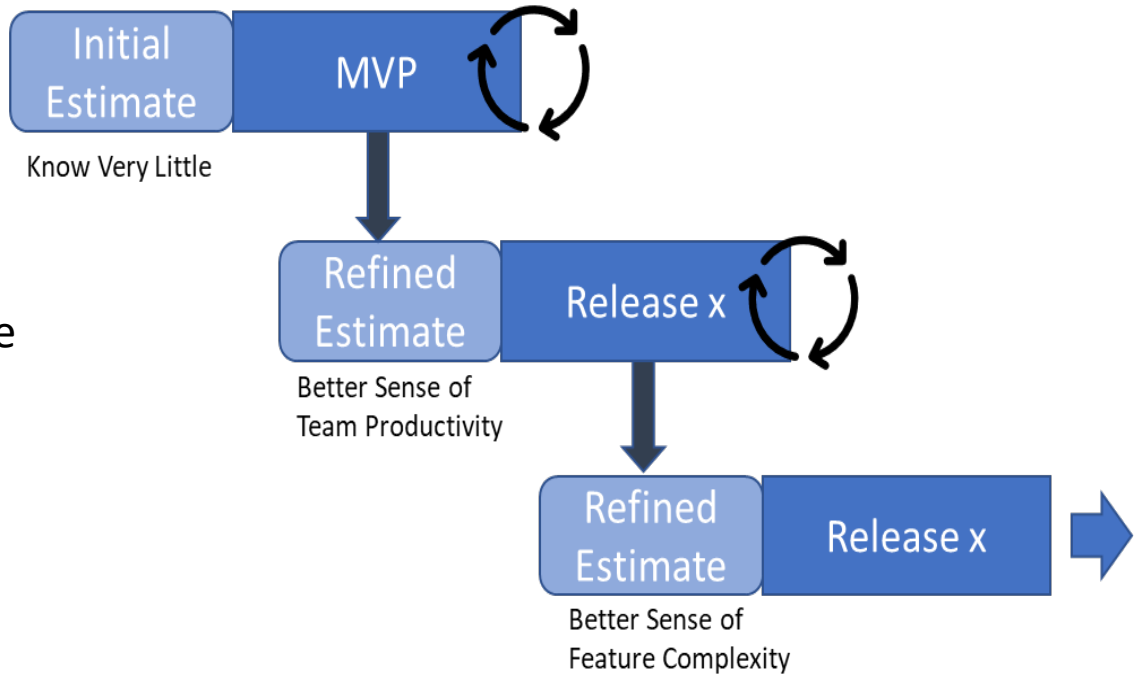
Overall Cost Estimate Drivers

To build into whatever methodology is being used

Areas to Explore	Cost Elements
- Planned Cadence to Achieve Roadmap - Size of the Software Development Effort - Complexity of the Desired Features - Likely Productivity of the Development Team	# of Software Teams to Achieve Roadmap
- Likely Productivity of the Development Team	Coaching Labor
- Team Organization (Single v Multiple; Consolidated v Geographically-separated)	Facility Costs SE/PM Costs
- Complexity of the Development Environment	Enterprise Services Licensing Costs
- Expected Level of User Training	Training Costs Tech Manual Development
- Level of Required Testing & Certification	Test and Eval Labor HW/SW for Mod/Sim Environment Interoperability Certification Labor

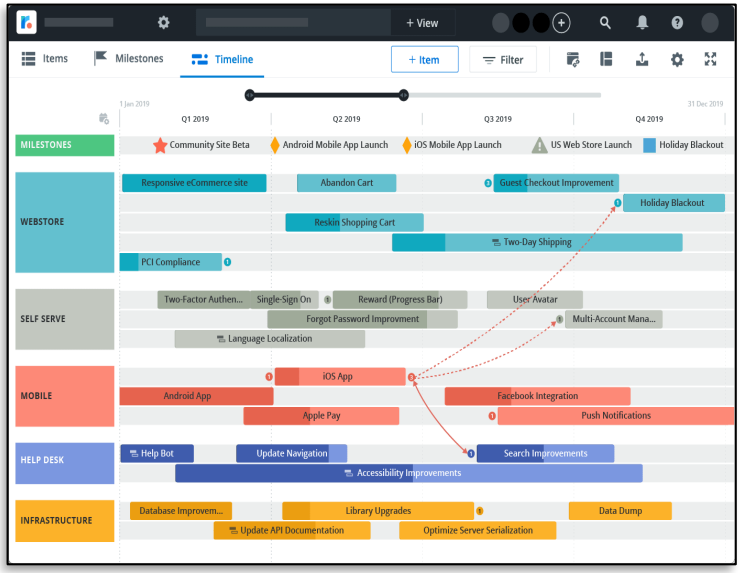
Use Data from Previous Releases

- The cost estimate should continually be updated as near-term releases complete and the next set of future releases becomes more defined.
- Understanding how much work the team can accomplish in each near-term release helps prioritize work as well understand the impacts of work delays.
- Data from completed releases can be collected early on and used to inform estimates for future releases.



Use the Program Roadmap

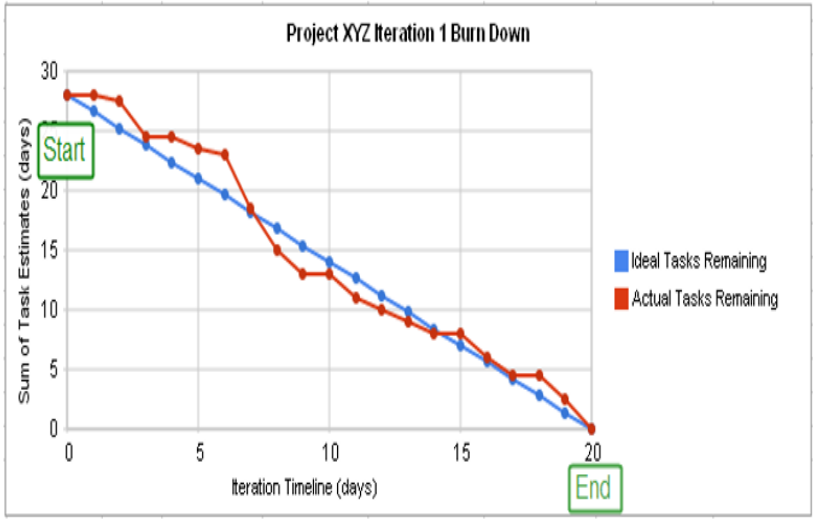
Based on the capabilities articulated in the CNS and user prioritization, the team should refine their initial decomposition into a Program Roadmap to support execution over the next 18 months to 2-year duration.



- **Roadmaps serve many useful purposes:**
  - **Keep the team connected to the higher-level objectives**
  - **Outlining a clear near-term path to achieve user's highest priorities**
  - **Facilitate cross-functional team collaboration**
  - **Channel of communication to the user community to set clear expectations.**
  - **Informs the Value Assessment**
- **For estimators, roadmaps should:**
  - **Allow for higher fidelity near-term estimates based on the clear goals**
  - **Serve as the basis for engaging with the development team to garner SME input (e.g. feature complexity)**
  - **Identify when the number of teams required to meet roadmap is not consistent with available resources**

Use Burn-Down Charts

- Burn Down charts track progress for each release aligned to the roadmap
- Show the amount of work on the project against time
- Can be used to inform estimate update cycles (as releases occur)
- Can also be used to help advocate for more resourcing



After Iteration x, your backlog is now = XX more story points remaining (some burn down and some additional inputs) and your demonstrated velocity is still = 15

Original Plan for remaining work = 8 weeks at $160K

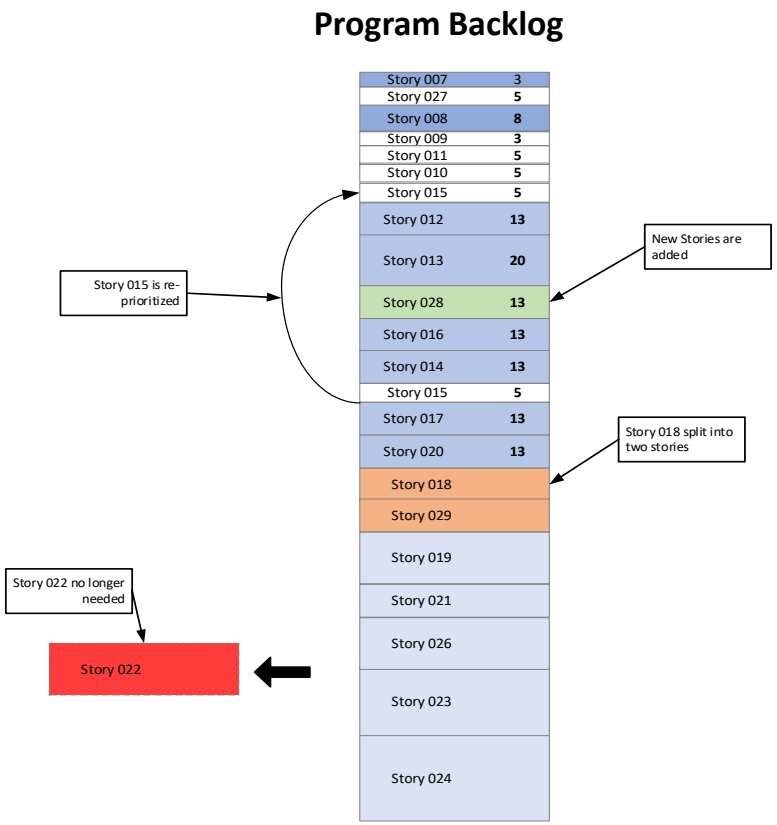
Revised estimate based on burndown = 12 weeks at $240K

Use Program Backlog Updates

The Product Backlog continually adapts to the needs of the users and drives updates to the Program Roadmap and eventually the CNS.

- New backlog items are added
- Backlog items are re-prioritized
- Backlog items are refined/split
- Some Backlog items are determined to be no longer needed

The Program Roadmap is updated based on these changes and subsequent user reprioritization.



Use Metrics

Avg Lead Time for Authority to Operate (days)	Average number of days to obtain authority to operate (ATO) by release
Mean Time to Resolve Experienced Cyber Event	Share insight into how quickly the program was able to resolve critical cyber events from the time of identification through resolution (provide average response time)
Mean Time to Experience Cyber Event	The mean time from time of cyber event start to time of identification
Avg Deployment Frequency	The average frequency of deployments into an operational environment. Select the value that aligns closest to the average frequency of deployments since last reported. Values are provided for selection (daily; weekly; semi-monthly; monthly; quarterly; annually; greater)
Avg Lead Time	The average duration time to deliver a capability or feature into operation, measured from the time the code is committed (development activity finished) to the time it is available for release to operations (production).
Minimum/Maximum Lead Time	The minimum and maximum duration times to deliver a capability or feature into operation, measured from the time the the code is committed (development activity finished) to the time it is available for release to operations (production).
Avg Cycle Time	The average duration time to deliver a capability or feature into operation, measured from the time the need is identified for a specific build (moved from the backlog to a planned release) to the time it is available for release to operations (production).
Change Fail Rate	The percentage of releases to the production/operational environment that requires subsequent remediation.
Mean Time to Restore	The mean time to restore the system in response to a downtime event or a defect that requires subsequent remediation.
Value Assessment Rating	The program office's perceived rating based on the feedback received from the operational sponsor.
Executive Summary from Last Value Assessment	Summary describing the user and program office's joint assessment of the value provided by the program from the previous period (at least annually but may be more often).

Minimum set of software pathway metrics that may provide useful to cost estimators in understanding development team performance.

Can Inform Risk Associated with:

- Productivity (deployment frequency)
- Efficiency (Lead and Cycle Time)
- Quality (Change Failure Rate)
- Security Posture (Cyber Events)
- User Perspective (Value Assessment)

Downward trends in these areas should impact risk buffer built into estimate refinements.

- Does the team need better tools?
- Is there technical debt to consider?
- Is there greater complexity than anticipated?

Use Value Assessments

To assess achieved increases to mission effectiveness

Measurement	Improvement Goal	Mission Effectiveness With New Features	Assessed Value
ID Range	From 50km to 70km	80km	Exceeded Goal. Can identify targets 10km farther than planned which increases engagement opportunities by
Accuracy	From 60% to 70%	80%	Exceeded Goal. 20% more reports accurate, reduced risk fratricide by x%
Operating Time	From 100 hours to 150 hours	150 hours	Met Goal. New software improves power utilization, an increases operating time
		Value Assessed	**High Value**

To assess value achieved in areas where the program office is less able to directly influence but where the quality of the product plays a major role.

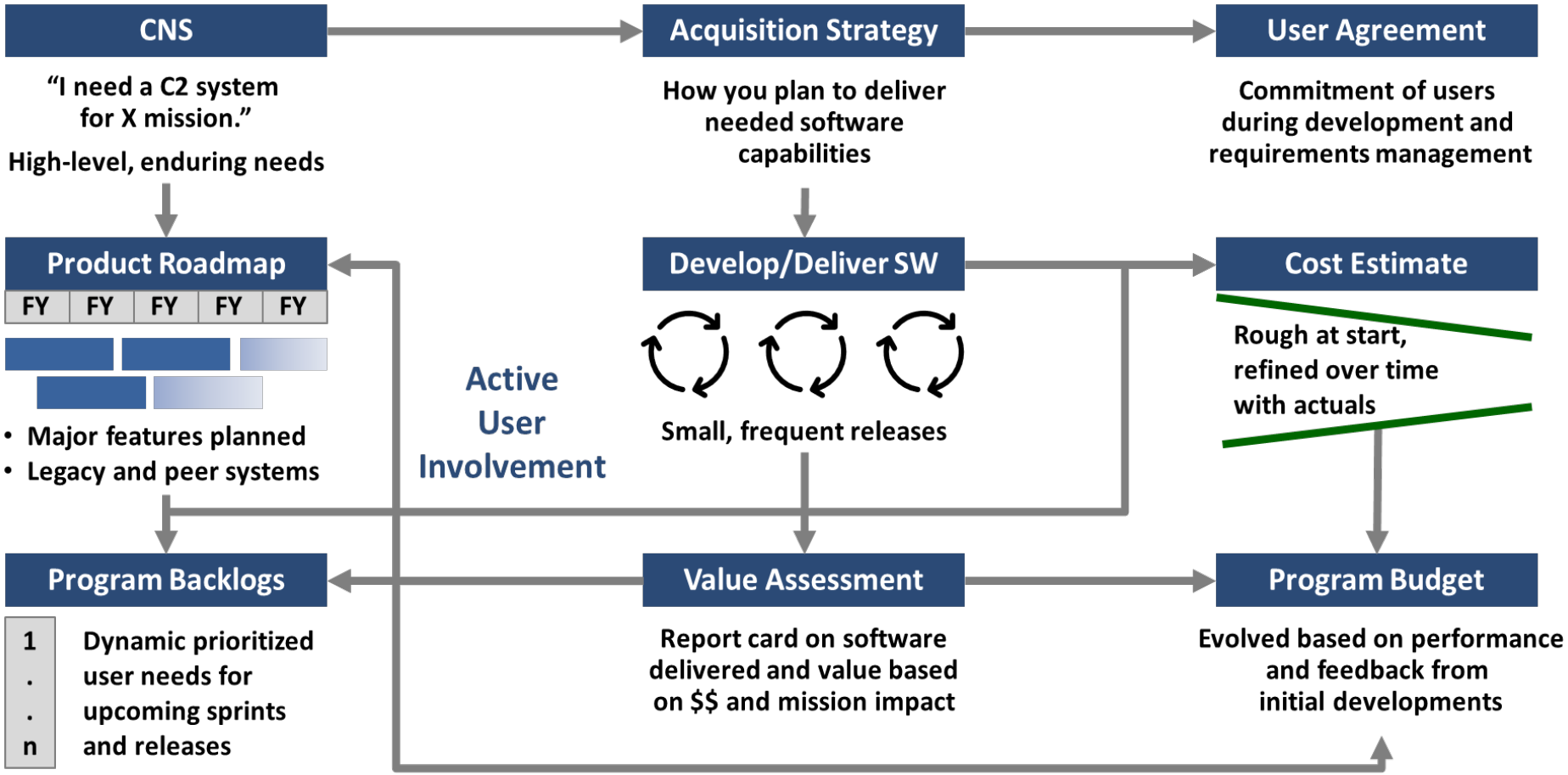
Measurement	Expected Level	Achieved Level	Assessed Value
User Adoption	Gain 100 users/month	Gained 500 users/month	The product developed for a wide set of users was deemed extremely useful and adoption is growing beyond expectation
		Value Assessed	**Exceptional Value**

Value Assessments can be a useful tool to understand if the program is properly resourced near-term based on current projections.

Highly positive feedback would confirm that the team is operating effectively and delivering on an acceptable deployment frequency

Highly negative feedback may drive some pivots in how the team is organized or staffed which could drive the need for more resources (even initially).

Pulling It All Together



Summary

- Continuous planning is critical for programs using modern software practices & cost estimation remains a critical activity to inform/defend appropriate resourcing.
- Cost estimators need to be as agile as the software program they support. They need to be integrated into the team's operations to gain the full understanding required since data deliverables may only show part of the picture.

- **Successful cost estimates**:
  - Combine team information and work to be delivered
  - Project costs by segments of work to be delivered (guided by a roadmap)
  - Consider complexity and incorporate past performance
  - Are updated regularly

Contact

Strengthening DoD Software Acquisition

BETTER

- High Mission Value
- Cyber Secure
- Enable Efficiencies

FASTER

- Lead Time – Need to Delivery
- Frequency of Releases
- Rapid Response to Operations/Cyber

CULTURE
Human-centered design, speed of delivery, and continuous improvement

POLICY
OSD, Joint Staff, and Service policies to provide flexible structure for modern software

PROCESS
Streamline and transform cost, requirements, T&E, cyber, and sustainment for software

TRAINING
Transform software training for DoD's acquisition and operational workforces

GUIDANCE
Provide how-to insights and resources to shape program strategies and execution

TOOLS
Leverage software factories, DevSecOps pipelines, enterprise platforms, services

Gen Hyten - Vice Chairman of the Joint Chiefs of Staff

- 2nd priority: "do everything I can to **insert speed** into the processes inside the Pentagon."
- Biggest thing we have to do in acquisition
  - allow people to take risk
  - give them the authority and responsibility
- the process that we have for building SW is horrible.

"What keeps me up at night is not North Korea, but that the U.S. has lost it's ability to go fast."
- Gen Hyten as STRATCOM Commander at AFA in 2017

https://www.csis.org/events/conversation-general-john-hyten-vice-chairman-joint-chiefs-staff

The DoD 4

1 Delivery Speed and Cadence

- Lead time; DORA
- Deployment frequency DORA

2 Stability & Reliability

- MTTR; DORA
- Deployment Failure Rate DORA

3 Value / ROI: DOD4+

- Value produced (and associated costs) – tailored to mission threads / stakeholders

4 Cyber Resilience DOD4+

- Time to detect/resolve cyber event; AVG time to achieve ATO

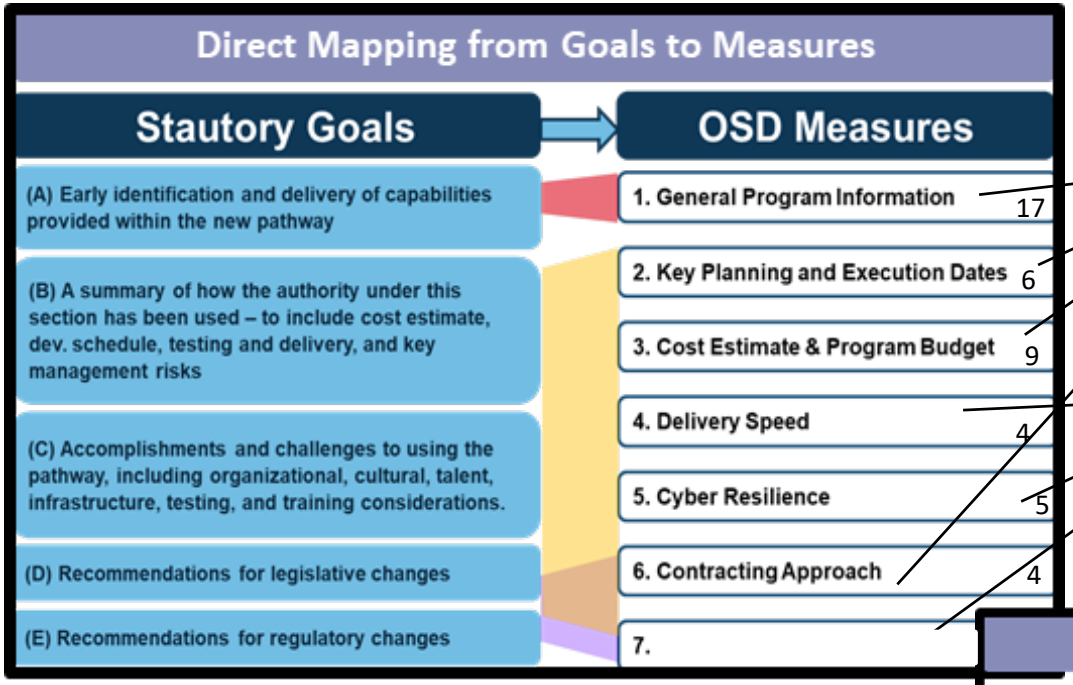
DOD4

E.g., Forms of Value

Time savings to execute a mission process
Increased accuracy of a provided solution that drives a mission decision
Personnel savings to execute a mission
Increased safety while executing a mission
Cost savings resulting from a software capability

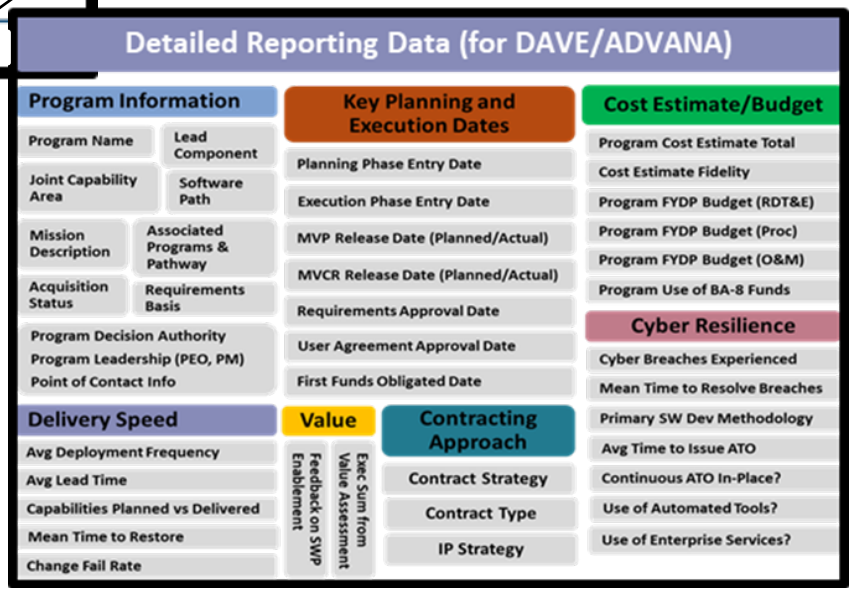
- **Product Performance & Maturity**
  - SW maturity/quality (defect backlog; defect resolution time; key -illities);
  - Product performance metrics and quality attributes *will be highly contextual*
- **Scale of Automation and Transformation**
  - Across product lines and mission threads: (% of product lines w/ build automation; % of tests cases automated)

PEO/PM (needs get more granular)



36 Uncontroversial / bare-min scoping/demographics to meet statute

12 Uncontroversial / bare-min performance to meet statute



<u>SWP Goal</u>: measuring performance of the SWP

- **MAXIMIZE SWP Insight** to improve it and reduce bureaucracy
- **MINIMZE the burden on programs**

Proactive data strategy & collection approach

- Closely integrated with the AVSG team
- Piloted with AF .87 program
- Feedback via OSD/Service SW Policy Alliance

Information Requirements

Balance speed with rigor – Focus on SW over extensive docs

Entering the Planning Phase	
ADM signed by DA	Draft CNS
Entering the Execution Phase	
Capability Needs Statement	User Agreement
Acquisition Strategy	Cybersecurity Strategy
Test Strategy	IP Strategy
Product Support Strategy	Information Support Plan
Program Cost Estimate	CARD
During the Execution Phase	
System Architecture	Product Roadmap
Program Backlogs	Strategy Updates
CARD/Cost Estimate Updates	Value Assessment
Metrics and Reporting	

See details at: https://aaf.dau.edu/aaf/software/develop-strategies/

Agile Drives Adjustments to Legacy CE Practices

Cost Estimation for Traditional Development	Component	Cost Estimation for Agile Development
Exhaustive upfront analysis; rigid baselines. Milestone driven updates	**Cost Estimating Process**	Iterative; integrated; collaborative through development cycles. Regular updates; cost analyst tightly coupled with teams
Solution defined in detail upfront. Update at program milestones	**Program Definition**	Near term iterations defined in detail with lower fidelity definition for longer term objectives. Update regularly between program milestones
Detailed elements structured by software development phase; clear life cycle phases and WBS breakout	**Work Breakdown Structure (WBS)**	Detailed elements by capability definition levels and iterations; blurred life cycle phases and WBS breakout
Large set of actual program data available; traditional software sizing metrics established	**Cost Data**	Emerging set of actual program data; challenges with leveraging data across programs. Actual data from early iterations can be used to inform future iterations. Native Agile software sizing metrics differ from established software metrics.
Uncertainty and risk focused on cost and schedule required to meet fixed scope	**Uncertainty and Risk Analysis**	Uncertainty and risk focused on variable capability and definition of done given cost and schedule
Trade-offs on solution details	**Trade-Offs**	Trade=offs on capability
No benefits are realized until end of development when software is fielded	**Benefit Analysis**	Benefits are realized faster with each fielded iteration of software

Estimating costs in an Agile environment requires an iterative, integrated, and collaborative approach

Challenge – No Standard Unit

- **There are different ways that software can be sized and measured**
  - Story Points / Features
  - Standard Component Analogy
  - T-Shirt Sizing
  - Simplified Function Points

- **<u>Key</u> is to Stay Consistent with the Program**
  - Maintain ability to compare projections to actuals
  - Maintain ability to make dynamic updates quickly
  - "Train" the team on the approaches – improve efficiency over time

Still a Lot of Learning on what Cost Estimating Practices Work Best When

Proposed SW Cost Estimating Principles

Nature of Agile Development

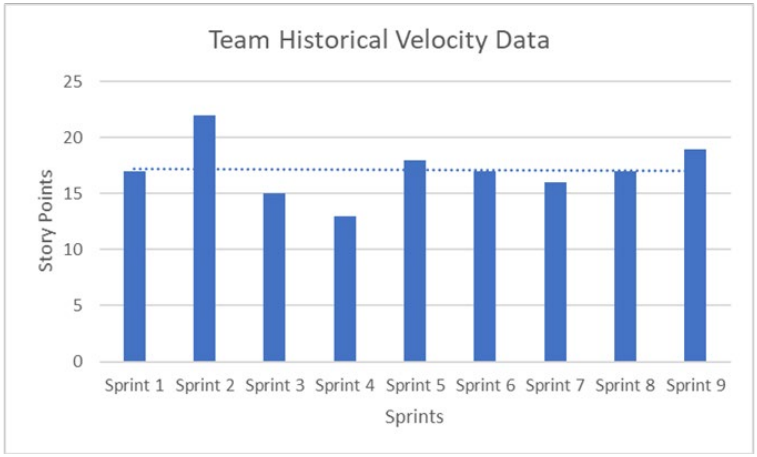
- Eschew complete definition of work scope up front
- Promote adherence to cost and schedule, flexing scope
- Assume a relatively constant pace of development, based on team steady state output and the number of teams employed
- Short term deliveries provide continual performance feedback to inform future estimates in execution phase
- Consider complexity as well as past performance in iteratively refining cost estimates

Cost Estimate Adjustments

- Accept a higher level of abstraction and less detail
- Use an iterative, integrated, and collaborative approach
- Employ capacity driven estimating methods (versus process driven)
- Update regularly, reflecting a balance of known vs unknown
- Use for planning and providing insights on what capability can be accomplished over time
- Cost out in segments aligned to the Product Roadmap

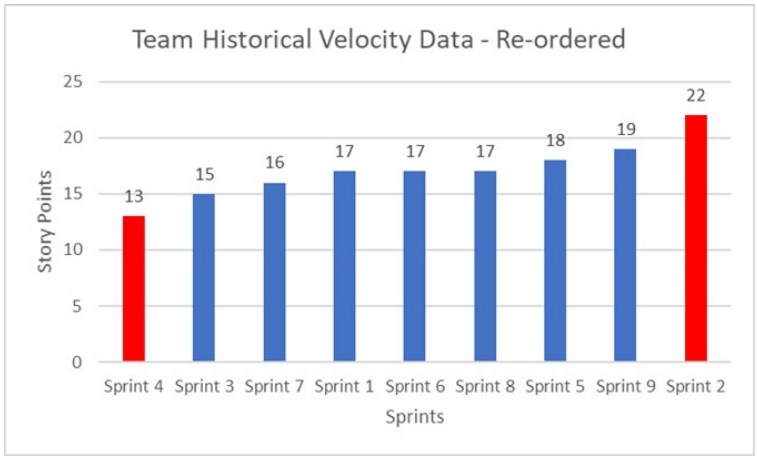
Use Historical Velocity to Refine Estimate

Monitor the Team's Historical Velocity



Determine a Confidence Interval

Identify outliers/deviations



Velocity = the amount of work the team can deliver in an iteration

- Measured as the sum of all the story/function points completed in an iteration
- Knowing prior team velocity assists with planning future work and helps ensure assigned work can be realistically accomplished
- Performance metrics may also play a helpful role in refining cost estimates.

Based on the analysis:

We are 90% confident that in the Development Team will continue to deliver somewhere between 15 to 19 Story Points per Sprint

As a team continues to execute the confidence interval will change. Continue to update as more historical data is collected.