National Defense Industrial Association
Integrated Program Management Division

Critical Path Management in an Incremental Product Delivery Environment White Paper

September 25, 2025

Revision 1.0

National Defense Industrial Association (NDIA)
2101 Wilson Blvd., Suite 700
Arlington, VA 22201
(703) 522-1820
Fax (703) 522-1885
www.ndia.org

© 2025 National Defense Industrial Association, Integrated Program Management Division (IPMD)

Permission to copy and distribute this document is hereby granted provided this notice is retained on all copies, copies are not altered, and the NDIA IPMD is credited when the material is used to form other copyrighted documents.

Table of Contents

1 Introduction 2
1.1 Purpose of the Guide 3
1.2 Acronyms and Terms 4
2 Product Planning Enables Work decomposition 6
2.1 Right-sizing Capabilities, Features, and Stories 6
2.2 Product Roadmap and Release Planning 7
2.3 Model the Product Roadmap and Release Plan in the IMS 10
3 Coping with Rapid Development and Changes 11
3.1 Keeping the IMS Current with PI scope 11
3.2 Acceptance and Communication of Change 11
3.3 Update and Communicate Critical Path 12

List of Figures

Figure 1: Work decomposition over time and level of backlog detail 7
Figure 2: Product Roadmap 8
Figure 3: Release Plan 9
Figure 4: Sequencing of capabilities and features in the IMS 10
Figure 5: Work Scope Change Log 12
Figure 6: Interim Milestone and Project Critical Path Analysis 13

© 2025 NDIA IPMD

1 Introduction

Program planning personnel using integrated schedules in a performance measurement environment, which may include an Earned Value Management System (EVMS), often have difficulty staying synchronized with Agile or iterative delivery work plans while having the proper fidelity for critical path visibility. Contributors to the lack of Critical Path visibility can be insufficient product planning and work decomposition. Additionally, programs may not always have visibility into the schedule impacts of program changes on critical and driving paths, due to the same aforementioned contributors.

There is a misconception that Critical Path analysis cannot be done on agile or iterative product delivery efforts. Critical Path management and reporting is feasible when an IMS is integrated with Agile Epics/Capabilities and Features, a quality task network is in place, and a disciplined scope change management process is implemented.

Prior to utilizing the application of critical path methodology with an agile methodology, readers are reminded of why critical path is important in iterative delivery. Key mechanisms by which integrating critical path and agile management improve product delivery transparency:

- *Prioritization of high-impact tasks:* Identifying the critical path highlights tasks whose delay would push back overall delivery. Agile teams, using tools and ceremonies such as sprint planning and daily stand-ups, focus efforts on these tasks, reducing the likelihood of delays and ensuring that valuable capabilities are delivered as soon as they are ready.
- *Continuous reassessment:* Agile methods involve regular review of progress and rapid adaptation to changes or issues. This flexibility ensures that changes in one area do not ripple unchecked into the critical path, allowing for contingency planning and quick pivots, minimizing wasted time.
- *Resource optimization:* By recognizing the most crucial, time-sensitive tasks, Agile teams can dynamically allocate specialized resources to those areas, reducing idle time and maximizing productivity across sprints or iterations.
- *Incremental value delivery:* Agile emphasizes breaking the program into smaller, manageable increments (e.g., user stories, features) that can be delivered and validated iteratively. Managing the critical path within this structure enables portions of the capability to be deployed faster rather than waiting for full project completion.
- *Bottleneck identification and mitigation:* By continuously visualizing dependencies and monitoring the critical path, teams can spot emerging bottlenecks early. Agile's iterative planning enables rapid action to resolve these, preventing later-stage project delays.

This document assumes a complete understanding of agile (agnostic of the specific agile framework) or iterative delivery methodologies, scheduling and EVM practices. This document will not address existing best practices or industry guidance regarding any specific agile methodology, scheduling or EVM. All existing critical path best practices and schedule risk assessment (SRA) practices are directly conveyed when utilizing an agile or iterative delivery methodology.

1.1 Purpose of the Guide

Program Managers have many processes at their disposal to utilize for project and program execution. Implementation and integration of the right processes contributes to performance measurement. This white paper offers guidance for managing Critical Path (CP) schedule analysis in an Incremental Product Delivery environment, including Agile and hybrid development models. It is intended for experienced program management and scheduling professionals who possess foundational knowledge in Agile, Critical Path Method (CPM), and Integrated Master Scheduling (IMS). Agile planning is disciplined with multiple levels of strategic visionary, interim releasable, and detailed short weekly planning. As work is reprioritized to complete the contract and deliver the scope, a logical order for work completion is required.

As more government programs are required to execute time-phased roadmaps and iterative execution models, confusion persists around how traditional CP management practices apply within Agile contexts. This paper demonstrates that when product scope is contractually defined and required to be delivered for contract completion, backlog items are properly decomposed, and IMS models align with the time-phased roadmap, Critical Path analysis is not only possible, but essential.

The objective is to align Agile delivery models with proven scheduling practices, emphasizing:

- Scope visibility must still align with contractual obligations
- Critical Path and Driving Path concepts remain valid regardless of delivery model
- Agile does not eliminate the need for disciplined planning, sequencing, or milestone integration

Ultimately, this paper aims to equip practitioners with a method that applies the same analytic rigor already used in traditional environments, to a schedule that is now mapped to a quality task network consisting of Agile constructs, based on the specific scope hierarchy. This paper references Capabilities and Features for a scope hierarchy example. Since the scope hierarchy may be different for each program or project, the content in this paper may be applied to alternate scope hierarchies.

1.2 Acronyms and Terms

Abbreviations and Acronyms:

CAM	Control Account Manager
EVM	Earned Value Management
IMS	Integrated Master Schedule
MVP	Minimum Viable Product
PI	Program or Planning Increment
SLPP	Summary Level Planning Package

Terms:

Agile - A flexible, iterative approach to project management that emphasizes incremental delivery, collaboration, and adaptability. It helps teams quickly respond to changing requirements while maintaining program objectives.

Backlog - A prioritized list of work items that need to be completed. It ensures teams focus on the most important tasks first while maintaining flexibility for shifting priorities.

Capability or Epic - A high-level major system function typically completed across multiple releases

Change Body - A formal organization that has the authority to implement change

Critical Path – Many organizations have varying definitions for critical path. A subset of those definitions include. NDIA IPMD PASEG: The continuous sequence of tasks which determines the program end date. These tasks prevent the program from finishing sooner. PMI (2021): The sequence of activities that represents the longest path through a project, which determines the shortest possible duration. DoD Earned Value Management System Interpretation Guide (EVMSIG), March 14, 2019, as follows: "A sequence of discrete work packages and planning packages (or lower-level tasks/activities) in the network that has the longest total duration with the least amount of total float/slack through an end point that is calculated by the schedule software application.

Demo - A review session where teams showcase completed work to stakeholders for feedback and validation. It helps to ensure the product meets requirements before full deployment.

Driving Path - The continuous sequence of tasks which determines the completion date of an interim program milestone/task. These tasks prevent the interim program milestone/task from finishing sooner.

Feature - A well-defined system function to be completed within a release

Interim Milestone - A milestone other than the end of the project that is used as a measure of incremental maturation or delivery of the entire project work scope. An MVP can be considered an interim milestone.

Major Release - Typically, major releases consist of significant packaged capability or functionality that is released to the end user after conducting stand-alone and integrated test. Releases should be defined and coordinated with all stakeholders on a program relative to their scope and purpose.

Pacing Activity – The earliest leading activity in either a Driving or Critical Path that is pushing all subsequent Critical/Driving Path work.

Planning Increment - A structured period defined by each project or program team (can be 8-12 weeks, or shorter) for teams to plan, develop, and integrate work. It aligns multiple teams on priorities, milestones, and risks. Can be referred to as a "release".

Product Planning - The process of defining, prioritizing, and scheduling product features. It ensures development efforts align with strategic objectives and operational needs. Throughout this paper, "product" planning is a generic reference and meant to be synonymous with release or program increment (PI) planning. Apply based on the specific terminology and methodology of the project or program.

Product Roadmap - A high-level timeline outlining planned features, milestones, and goals. It provides a long-term vision to guide development and resource planning.

Program Planning and Controls Personnel – Individuals who conduct planning and tracking of work

Release Plan - An agreed-to plan which states what is in and out of scope by release. Scope may be defined at various levels in the Release Plan.

Story - A small and well-defined implementation detail that contributes to the completion of the feature system function within one sprint

2 Product Planning Enables Work decomposition

2.1 Right-sizing Capabilities, Features, and Stories

Government contracts typically require target scope completion regardless of delivery method; therefore, it is typical to include all contact scope in the high-level program roadmap and incrementally decompose the scope to lower levels during execution. When developing a right-sized backlog, it is essential to understand the required technical goals for a particular “release” cycle, whether the release is internal to test on a more frequent basis, possibly at each Planning Increment, or to the end user on a less frequent basis at each major release, consistent with the customer desired release schedule. Recognize that technical goals and maturity often emerge through iteration (over multiple sprints and releases) and learning. Consider integrating backlog refinement practices that evolve scope maturity over time as synchronized with the release goals. The estimated scope, consistent with agile forecasting techniques, required to meet that maturity must be validated with available resources/velocity and estimated story size to confirm scope completion and planned duration is achievable within each Planning Increment (see Figure 1 below).

Due to the specifics of the program, some program teams may not initially have enough information to be able to identify and/or prioritize work beyond near-term Planning Increments. If the work is on a completion-effort contract with 'must do' work to satisfy contractual terms, then it is typical that an Integrated Master Schedule (IMS) will be planned and scheduled to both interim and end objectives, consistent with adaptive backlog planning. If the program's prioritized backlog only extends into the near term, care must be taken to ensure that the Roadmap and the IMS contain adequate inputs to support all contractual goals. Consistent with the agile practice of progressive elaboration, the Planner may or may not have sufficient decomposed inputs to support the full long-term plan to meet all end objectives. To represent the future vision of scope within the contract, programs should continuously apply “Multiple Horizons of Planning” (NDIA Adapt Paper IEEE 15288 Meets Lean Agile, https://www.ndia.org/-/media/sites/ndia/divisions/ieee-15288-meets-lean-agile_v12.pdf?download=1) to size, estimate and address the uncertainties in the prioritized backlog at higher levels of elaboration in the scope hierarchy. As such, planning packages and/or Summary Level Planning Packages (SLPPs) should have a relationship to the scope and Roadmap (see Figure 2) to support longer range planning.

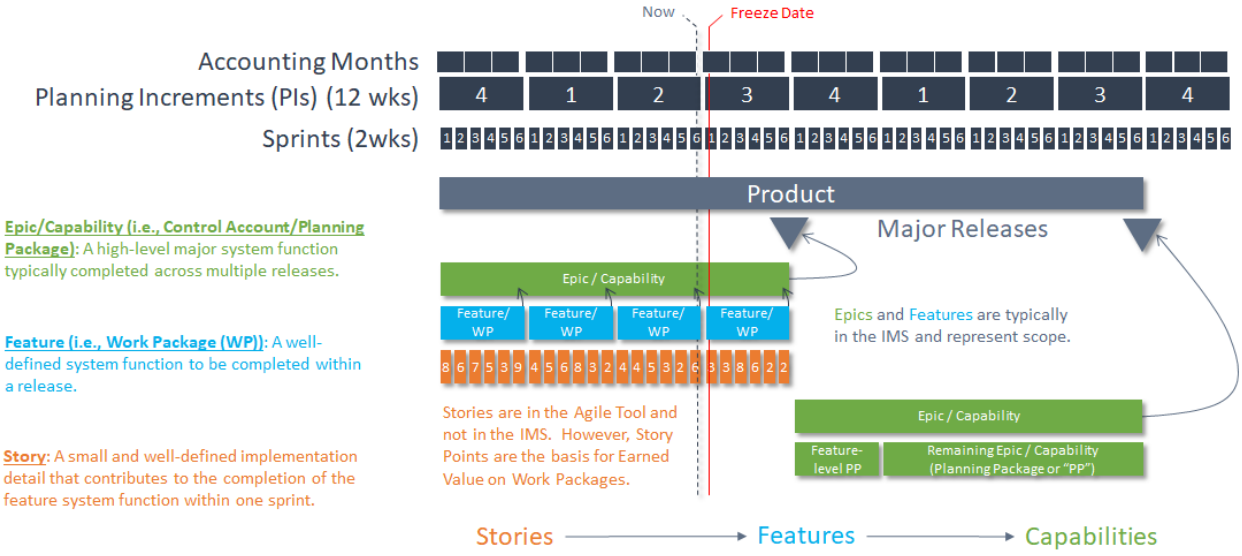


Figure 1: Work decomposition over time and level of backlog detail

2.2 Product Roadmap and Release Planning

The Product Roadmap (see Figure 2 below) helps visualize the buildup of technical maturity leading to product delivery. The Product Roadmap is a contractual planning tool that depicts phasing of Capability, select key Features, Demos, and integration points with any stakeholder on or off-program. External dependencies that are agile or non-agile may be incorporated in the Product Roadmap and will be represented in the IMS.

Dependencies may occur at various levels within the Backlog. However, typically, more significant dependencies at the Feature or Capability level are modeled in the Product Roadmap and IMS. Keep in mind that all dependencies should be captured, as necessary, and modeled appropriately in the Backlog down to the Story level.

Product Roadmaps may vary in scope and product hierarchy. Smaller programs or projects may choose to have one page to reflect their vision. Other, more complex, programs may have several roadmaps to reflect different major releases, customers, or types of products. In addition to the standard agile practice of testing within the capability or feature development, there may be additional internal testing milestones reflected in roadmap. Adapt the roadmap to specific contractual obligations, as needed, as figure 2 is one example.

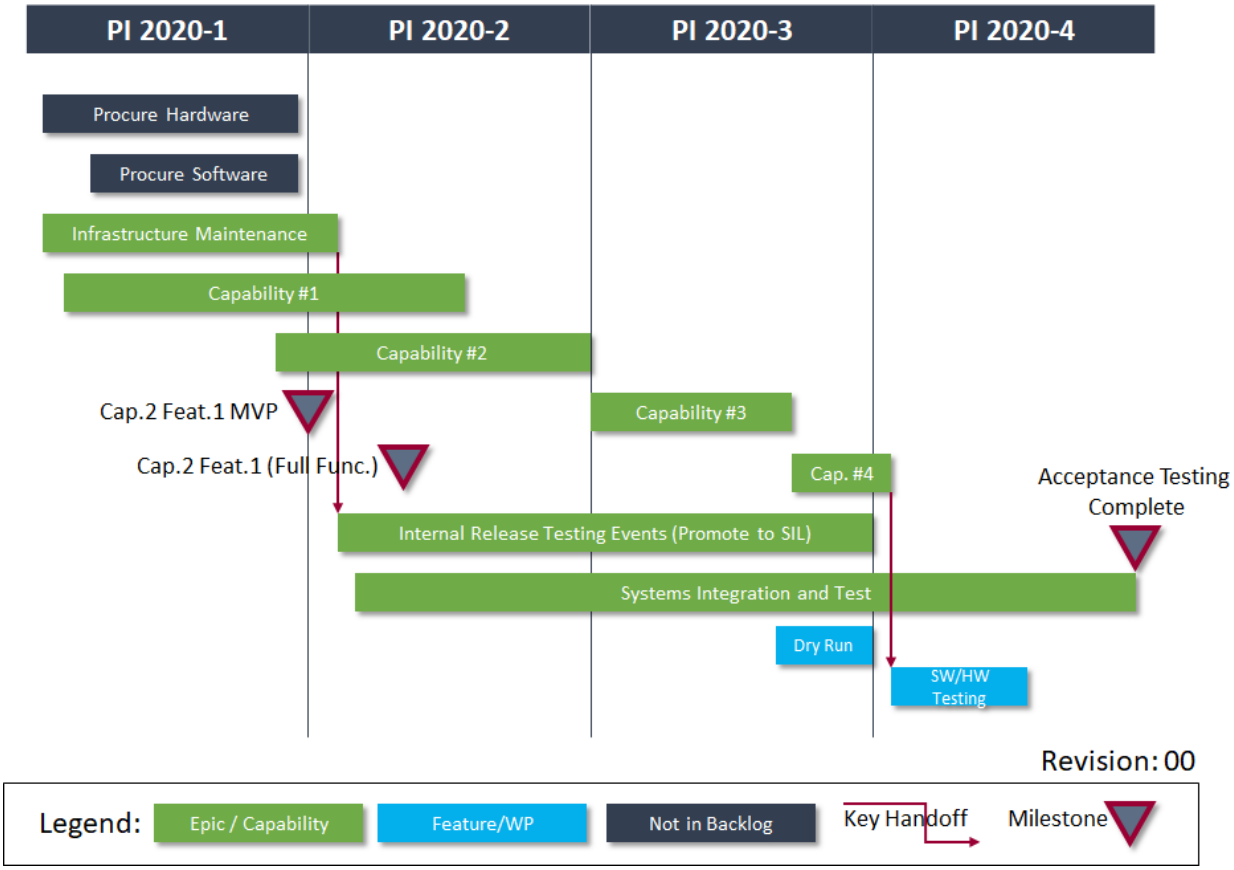


Figure 2: Product Roadmap

A configuration-controlled Release Plan (see Figure 3 below) should be created along with the Product Roadmap and high-level backlog content (typically at the Epic or Capability level) in agreement with the Systems Engineering process. The end user should be highly engaged with the Product Roadmap, organized around value, and Release Plan content with set priorities and objective evaluation criteria.

Once the high-level backlog content is deemed executable by all stakeholders, Interim Milestones and Demos, which relate to Features, should be incorporated into the Product Roadmap, consistent with the determined agile cadence. This allows the end user to stay engaged throughout product development which minimizes surprises when final product delivery occurs.

On very complex work, the end user and team may opt to demo or deliver a subset of capability or a Minimum Viable Product (MVP) with the intent to develop a fully functional product later. In those cases, ensure the MVP is labeled as such and accounted for later in the Release Plan.

Before finalizing lower-level plans (Features and Stories), they should be compared to available resources/velocity and estimated story size just as Capabilities and Epics were in the high-level plan.

© 2025 NDIA IPMD

2020			
Planning Increment /Release 1	**Planning Increment /Release 2**	**Planning Increment /Release 3**	**Planning Increment /Release 4**
Establish Common Software Development Environment (E-00024)			
Cap 1 - Feature #1 (E-00041)		Capability #3 (E-00048)	
Cap 1 - Feature #2 (E-00042)			Capability #4 (E-00049)
Cap 1 - Feature #3 (E-00043)			
Cap 2 - Feature #1 (E-00045) - MVP	Cap 2 - Feature #1 (E-00046) - Full Functionality		
	Cap 2 - Feature #2 (E-00047)		

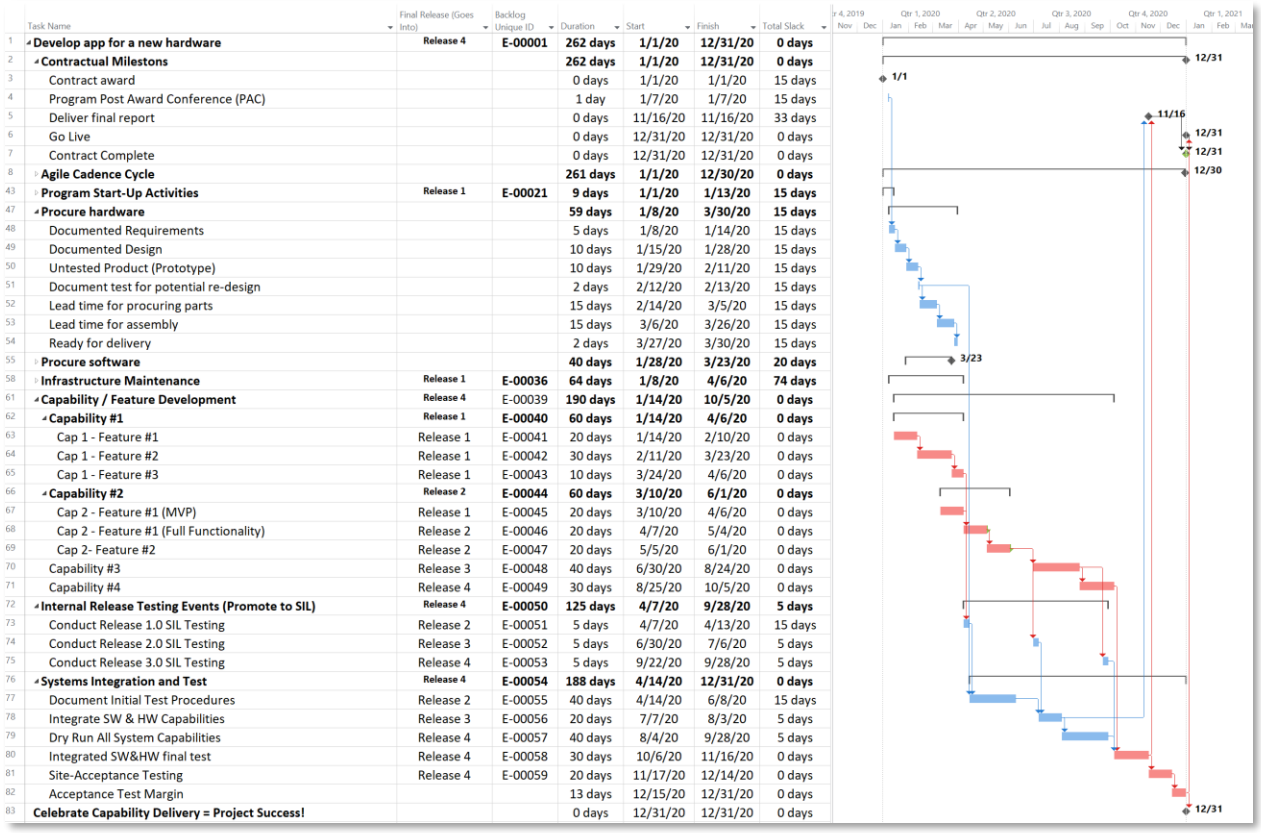
Revision: 00

Figure 3: Release Plan

2.3 Model the Product Roadmap and Release Plan in the IMS

As the Product Roadmap and Release Plan are developed, it is a good idea to start developing the IMS (see Figure 4 below) at a similar level and flow, capturing key handoffs as dependencies between Features and Capabilities in multiple planning horizons. This should include decomposition at more than current PI or PI+1 ideally through MVP. In cases where Planning Packages, at the Control Account level, are not identifiable, SLPP, at a higher level, may be used to cover future years within the Period of Performance. As with any scheduling best practice and SLPPs EVMSD policy, decompose the SLPP as soon as practical, thereby increasing the quality of the Critical Path analysis over the program lifecycle. SLPP decomposition will be consistent with planning events and backlog grooming.

On near-term plans, when developing Stories in the Backlog, some key Story-to-Story dependencies may be useful dependencies to model between Features but without adding Stories to the IMS as line-items. Features can be used to model Story-to-Story dependencies because dependencies can be considered met when Feature is complete. Story completion in Agile does not make full functionality available until all Feature acceptance criteria are met.



	Task Name	Final Release (Goes Into)	Backlog Unique ID	Duration	Start	Finish	Total Slack
1	**Develop app for a new hardware**	**Release 4**	**E-00001**	**262 days**	**1/1/20**	**12/31/20**	**0 days**
2	**Contractual Milestons**			**262 days**	**1/1/20**	**12/31/20**	**0 days**
3	Contract award			0 days	1/1/20	1/1/20	15 days
4	Program Post Award Conference (PAC)			1 day	1/7/20	1/7/20	15 days
5	Deliver final report			0 days	11/16/20	11/16/20	33 days
6	Go Live			0 days	12/31/20	12/31/20	0 days
7	Contract Complete			0 days	12/31/20	12/31/20	0 days
8	**Agile Cadence Cycle**			**261 days**	**1/1/20**	**12/30/20**	**0 days**
43	**Program Start-Up Activities**	**Release 1**	**E-00021**	**9 days**	**1/1/20**	**1/13/20**	**15 days**
47	**Procure hardware**			**59 days**	**1/8/20**	**3/30/20**	**15 days**
48	Documented Requirements			5 days	1/8/20	1/14/20	15 days
49	Documented Design			10 days	1/15/20	1/28/20	15 days
50	Untested Product (Prototype)			10 days	1/29/20	2/11/20	15 days
51	Document test for potential re-design			2 days	2/12/20	2/13/20	15 days
52	Lead time for procuring parts			15 days	2/14/20	3/5/20	15 days
53	Lead time for assembly			15 days	3/6/20	3/26/20	15 days
54	Ready for delivery			2 days	3/27/20	3/30/20	15 days
55	**Procure software**			**40 days**	**1/28/20**	**3/23/20**	**20 days**
58	**Infrastructure Maintenance**	**Release 1**	**E-00036**	**64 days**	**1/8/20**	**4/6/20**	**74 days**
61	**Capability / Feature Development**	**Release 4**	E-00039	**190 days**	**1/14/20**	**10/5/20**	**0 days**
62	**Capability #1**	**Release 1**	**E-00040**	**60 days**	**1/14/20**	**4/6/20**	**0 days**
63	Cap 1 - Feature #1	Release 1	E-00041	20 days	1/14/20	2/10/20	0 days
64	Cap 1 - Feature #2	Release 1	E-00042	30 days	2/11/20	3/23/20	0 days
65	Cap 1 - Feature #3	Release 1	E-00043	10 days	3/24/20	4/6/20	0 days
66	**Capability #2**	**Release 2**	**E-00044**	**60 days**	**3/10/20**	**6/1/20**	**0 days**
67	Cap 2 - Feature #1 (MVP)	Release 1	E-00045	20 days	3/10/20	4/6/20	0 days
68	Cap 2 - Feature #1 (Full Functionality)	Release 2	E-00046	20 days	4/7/20	5/4/20	0 days
69	Cap 2- Feature #2	Release 2	E-00047	20 days	5/5/20	6/1/20	0 days
70	Capability #3	Release 3	E-00048	40 days	6/30/20	8/24/20	0 days
71	Capability #4	Release 4	E-00049	30 days	8/25/20	10/5/20	0 days
72	**Internal Release Testing Events (Promote to SIL)**	**Release 4**	**E-00050**	**125 days**	**4/7/20**	**9/28/20**	**5 days**
73	Conduct Release 1.0 SIL Testing	Release 2	E-00051	5 days	4/7/20	4/13/20	15 days
74	Conduct Release 2.0 SIL Testing	Release 3	E-00052	5 days	6/30/20	7/6/20	5 days
75	Conduct Release 3.0 SIL Testing	Release 4	E-00053	5 days	9/22/20	9/28/20	5 days
76	**Systems Integration and Test**	**Release 4**	**E-00054**	**188 days**	**4/14/20**	**12/31/20**	**0 days**
77	Document Initial Test Procedures	Release 2	E-00055	40 days	4/14/20	6/8/20	15 days
78	Integrate SW & HW Capabilities	Release 3	E-00056	20 days	7/7/20	8/3/20	5 days
79	Dry Run All System Capabilities	Release 4	E-00057	40 days	8/4/20	9/28/20	5 days
80	Integrated SW&HW final test	Release 4	E-00058	30 days	10/6/20	11/16/20	0 days
81	Site-Acceptance Testing	Release 4	E-00059	20 days	11/17/20	12/14/20	0 days
82	Acceptance Test Margin			13 days	12/15/20	12/31/20	0 days
83	**Celebrate Capability Delivery = Project Success!**			0 days	12/31/20	12/31/20	0 days

Figure 4: Sequencing of capabilities and features in the IMS

3 Coping with Rapid Development and Changes

3.1 Keeping the IMS Current with PI scope

Before an IMS is created, the Program Planner should meet with the Agile team, Program or Project Manager, and related CAMs to discuss operating rhythm and rules of engagement regarding planning, integrating, and reporting of the IMS in relation to the Agile processes. A reminder that release or program increment planning will be synchronized with rolling wave planning. Consider utilizing user defined fields in either the IMS (feature ID), or agile tool (WP ID) for direct horizontal traceability between the schedule and the agile tool.

Prior to PI Planning Event

Prior to the event, the Program Planner should posture the IMS to align with approved Features for the upcoming PI. In addition, ensure that non-approved Features and near-term non-decomposed Capabilities are far enough out in the future (past the current and possibly current +1 PIs). To accomplish this, changes to the Product Roadmap and Release Plan need to be monitored and modeled in the IMS. Having a clean list of Features entering the PI allows teams to focus on Story development and low-level dependencies rather than shuffling Features.

During PI Planning Event

During the event, the Program Planner should be available to advise PI Planning attendees of impacts on the IMS for certain changes that are discussed. The program schedule may be used to create what-if schedules and perform SRAs that model those changes and show impacts to the Program Milestones and resources. The output of the PI Planning Event will be the approved backlog, Roadmap and Release Plan to complete the feature scope for the next increment.

After PI Planning Event

After the event, the Program Planner should seek to determine if any changes are made to the backlog impacting phasing of Features or Capabilities. The Product Roadmap and Release Plan should be referenced again at this point. Those changes should be incorporated into the IMS as soon as possible as many times financial reporting periods follow shortly after PI events. Again, try to leverage Agile Change Body decisions for making Contractual, Financial and IMS changes. This may reduce the incorporation time into the EVM system.

3.2 Acceptance and Communication of Change

Before the Product Roadmap, Release Plan, Features, IMS and corresponding tasks or Work Packages are placed under configuration control, there should be a Change Body established in the configuration management plan (including roles and responsibilities) and contract language that approves change of technical work scope. The end user or customer must be part of this process and drive decisions on priorities and work scope. This Change Body will streamline the change management process.

There are various ways to document how work is turned on, off, or changed. The Change Body should select a mechanism (perhaps a tool and process, a database, or log) to track scope changes. This can be a sheet or database that records scope changes and logs change approvals. Contracts personnel for both the customer and the contractor must be engaged as formal letters are still likely needed.

When technical scope changes are made through the Agile execution process and Change Board, those decisions should be used as authority to implement change in our EVM and IMS systems (correlate scope changes seamlessly to task and work package changes in accordance with the contractor's EVMSD, which aligns with EIA-748 guidelines). Sometimes, program processes account for change management through multiple boards. As this approach is not streamlined, leverage change decisions already made and documented in the Agile process and tools so redundant approvals are not needed in other approval bodies, functions, and/or tools. Structure one Change Board to include all functions required to satisfy one integrated event.

Once a change is formally approved, that change should be communicated to all impacted parties, as defined by the Change Body, via a method that is easy to understand the nature and impact of the changes (see Figure 5 below). The format will vary widely but the intent is to have a log used by all stakeholders.

Work Scope Change Log

Contract Reference	Change	Approval Date	Notes
Contract XYZ. CLIN0001.	Fund necessary procurements for software and hardware (for entire project) as well as labor for setup of the Software Development Environment (SWDE) and other lab infrastructure. Includes Capabilities 1 and 2 for PIs 1 and 2 in CY 2020 (including necessary SIL and SIT activities).	12/31/2019 (Actual)	Award fee tied to successful MVP in PI1 on Feature 1 for Capability 2.
Contract XYZ. CLIN0002.	Fund Capabilities 3 and 4 as well as necessary SIL and SIT activities - including Acceptance Testing.	4/30/2020 (Actual)	Discuss PI3 readiness and remaining work scope in April PMR.
Contract XYZ. CLIN0001.	Originally planned for requirements 950-1000 to be satisfied in capability 5. Requirements 950-1000 were satisfied in capability 2 by implementing requirements 100-140. Removing requirements 950-1000 without impact to contractual scope or budget.	5/15/2020 (Actual)	Specific implementation had alternative ways to satisfy all requirements.

Revision: 00

Figure 5: Work Scope Change Log

3.3 Update and Communicate Critical Path

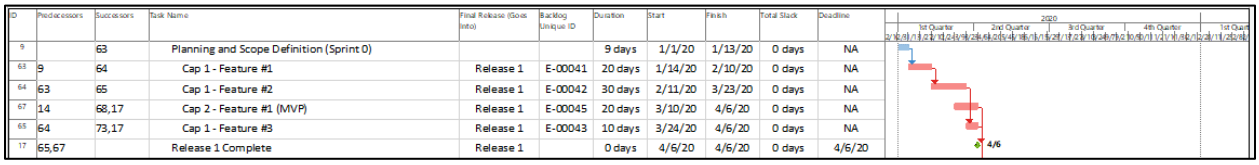
Once changes have been incorporated in the IMS following PI Planning, Total Float and Critical Path Analysis can be finalized and published to the team. Utilize existing analytic approaches (including schedule risk assessments (SRA) and tools as a scheduling best practice, as the backlog content from the approved release plan is synonymous with any other IMS "task"– regardless of delivery methodology. Because of the planning approaches described within this paper, a feature in an IMS with an agile delivery methodology, is now equivalent to any IMS task or work package that contains scope in a "non-agile" delivery method. Successful schedule updates prior to PI Planning and PI Planning engagement by the Program Planner reduce Critical Path surprises.

It is typical for the tasks and/or features which make up the Program Critical path to change as planning increments occur. This is because any change in forecast dates by way of statusing

the schedule, and any change in prioritization of Features (IMS tasks and logic), could have an impact on program driving and critical path calculations.

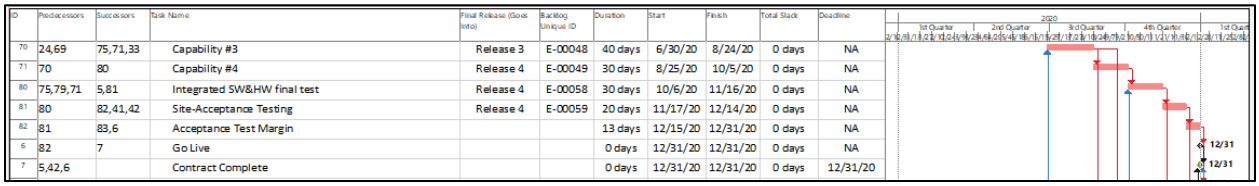
For example, an interpretation of figure 6 may be: release 1 is high risk; potential mitigation includes re-planning Cap 1 Feature 3 delivery to the next release, moving Cap 1 Feature 2 MVP start date earlier, if possible, and possibly minimizing Capability 4 scope to reduce the pressure on the acceptance test margin. Risks and opportunities are assessed as execution continues as part of managing the path to complete the delivery of the product.

Lowest Float / Driving Path to Release 1 (Interim Milestone):



ID	Predecessors	Successors	Task Name	Final Release (Goes Into)	Backlog Unique ID	Duration	Start	Finish	Total Slack	Deadline
9		63	Planning and Scope Definition (Sprint 0)			9 days	1/1/20	1/13/20	0 days	NA
63	9	64	Cap 1 - Feature #1	Release 1	E-00041	20 days	1/14/20	2/10/20	0 days	NA
64	63	65	Cap 1 - Feature #2	Release 1	E-00042	30 days	2/11/20	3/23/20	0 days	NA
67	14	68,17	Cap 2 - Feature #1 (MVP)	Release 1	E-00045	20 days	3/10/20	4/6/20	0 days	NA
65	64	73,17	Cap 1 - Feature #3	Release 1	E-00043	10 days	3/24/20	4/6/20	0 days	NA
17	65,67		Release 1 Complete	Release 1		0 days	4/6/20	4/6/20	0 days	4/6/20

Critical Path to Contract / Project Complete (future start due to R3 resource constraint)**:**



ID	Predecessors	Successors	Task Name	Final Release (Goes Into)	Backlog Unique ID	Duration	Start	Finish	Total Slack	Deadline
70	24,69	75,71,33	Capability #3	Release 3	E-00048	40 days	6/30/20	8/24/20	0 days	NA
71	70	80	Capability #4	Release 4	E-00049	30 days	8/25/20	10/5/20	0 days	NA
80	75,79,71	5,81	Integrated SW&HW final test	Release 4	E-00058	30 days	10/6/20	11/16/20	0 days	NA
81	80	82,41,42	Site-Acceptance Testing	Release 4	E-00059	20 days	11/17/20	12/14/20	0 days	NA
82	81	83,6	Acceptance Test Margin			13 days	12/15/20	12/31/20	0 days	NA
6	82	7	Go Live			0 days	12/31/20	12/31/20	0 days	NA
7	5,42,6		Contract Complete			0 days	12/31/20	12/31/20	0 days	12/31/20

Figure 6: Interim Milestone and Project Critical Path Analysis

Critical Path management and reporting is enabled when an IMS is integrated with Epics/Capabilities and Features, a quality task network across multiple planning horizons is in place, and a disciplined scope change management process is implemented.