SYM-AM-21-081

# Excerpt from the Proceedings

of the

# Eighteenth Annual Acquisition Research Symposium

## Agile Improvements to Critical Path Method

**May 11–13, 2021**

**Published: May 10, 2021**

Approved for public release; distribution is unlimited.

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

Disclaimer: The views represented in this report are those of the author and do not reflect the official policy position of the Navy, the Department of Defense, or the federal government.

Acquisition Research Program
Graduate School of Defense Management
Naval Postgraduate School

The research presented in this report was supported by the Acquisition Research Program of the Graduate School of Defense Management at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

# Agile Improvements to Critical Path Method

**Caitlin Kenney** —is a Systems Engineer and Agile Coach for International Systems Management Corp. She has an MS in Systems Engineering from University of Maryland and a BS in Industrial Engineering from Northeastern University. Kenney is currently pursuing a PhD in Civil Engineering, Project Management at University of Maryland. She has over 10 years of experience supporting DoD projects, including two DoD Agile Pilot Programs and multiple agile project management adoptions. She is a licensed professional engineer in Illinois and Maryland. [Ckenney7@gmail.com]

## Abstract

Agile project managers have a difficult time communicating schedules in traditional formats. For example, integrated master schedules, work breakdown structures, and so on are expected by project leadership, especially if they are new to agile. Agile does not include detailed level planning, years in advance. Instead, it focuses on small increments and iterates on the scope until the stakeholder need is met. Detailed planning is only done within the confines of an increment. Higher level milestones are captured for future efforts, but the detailed level planning is not done until nearing start of the work. Limitations of traditional project management techniques are nothing new. However, management of such projects still has not changed or significantly improved upon monitoring and controlling methods since their inception. The traditional Critical Path Method (CPM) is an example. CPM has the following known limitations. There is an assumption that durations can be accurately estimated and forecast. Estimates are based on historical data and made by separate planning departments, not the “doers”—the people closest to the work. Interdependencies are not always clear up front—they evolve as more information becomes available and work is further broken down, making it difficult to differentiate between scope creep and scope refinement. This can result in lengthy and costly contractual battles between the contractor and developer. Schedule logic in CPM is largely driven by finish-to-start dependencies, making other types of dependencies difficult to forecast and the critical path difficult to identify without software. Software requires user training/results in a designated “scheduler,” removed from the work. Management assumes projects can add resources to speed up completion, failing to account for learning curves, increased complexity, and additional bottlenecks. By applying production management and agile principles, projects can improve on the critical path method and develop specific agile applications.

**Keywords**: agile, software, cpm, production management

## Introduction

Traditional project management processes aimed to monitor and control cost, schedule, and performance are based on rigid estimating techniques applied to a single baseline early in a project life cycle. An agile approach allows for rolling-wave planning and execution, resulting in decreased variability and improved forecasting. This paper explores how agile principles of iterative and incremental planning and execution applied to traditional monitoring techniques for cost and schedule, such as Critical Path Method (CPM) scheduling can benefit from an iterative, agile approach. When projects are treated as a system, planning and execution are not siloed to respective parties; production design and processes are better understood, leading to improved capacity-based planning, reduced variability, reduced work in process, and increased throughput. ISE and agile share many common principles and methodologies. Applying these to project management results in improved monitoring and control.

### Research Issue Statement

How can projects use CPM in an agile project to create a detailed enough plan for project managers? Forecasting is only as good as the data you put in. However, if projects adopt an agile approach to CPM, projects can leverage the benefits of iterative, incremental develop to forecast progress and completion. Many traditional project management organizations that adopt agile try to impose traditional methods of control (CPM, Earned Value Management [EVM], etc.). They can be applied, but they must account for the iterative approach of agile. Applying an agile approach to CPM could improve on the method and provide agile project managers another means of measuring and presenting schedule data.

### Research Results Statement

A deeper understanding of the process design, production design, capacity planning, inventory control, and variability is needed in order to improve CPM. Production process and design should be used to develop the scheduling model accounting for capacity of resources (people, equipment, etc.). Capacity-based planning and velocity-based planning need to be the drivers behind the scheduling duration estimates (rates drive dates). By adopting agile, projects can adapt the CPM approach to account for rolling-wave planning, iterative development, and incremental changes, resulting in improved project performance.

## CPM Background

The critical path method (CPM) is defined as “a method used to estimate the minimum project direction and determine the amount of schedule flexibility on the logical network paths within the schedule model” (Project Management Institute [PMI], 2017b). CPM uses a schedule network analysis technique, independent of resources, to calculate the estimated project duration based on early start/finish and late start/finish activity dates. A forward and backward pass is conducted to determine the earliest start/finish and late start/finish based on the project activity estimated durations and sequencing. By creating a network diagram of activities based on predecessor and successor requirements and estimated task duration, the critical path and float (flexibility) of a project can be estimated (PMI, 2017a).

### Schedule Logic and Precedence Diagraming

There are four types of logical relationships and dependencies in CPM that affect the project schedule duration. The following definitions summarize the relationships (PMI, 2017a):

- **Finish-to-start (FS):** predecessor activity must finish before successor activity can start
- **Finish-to-finish (FF):** successor activity cannot finish until a predecessor activity has also finished
- **Start-to-start (SS):** successor activity cannot start until predecessor activity has also started
- **Start-to-finish (SF):** successor activity must start before predecessor activity can finish

In a schedule network diagram, an activity node will contain the activity name, early start, early finish, late start, late finish, total float, and duration (see Figure 1).

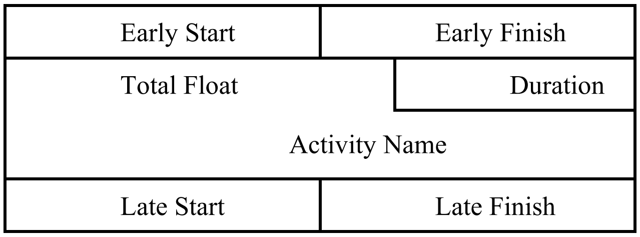

Figure 1. Activity Node in CPM Network Diagram

### Float

Float, or slack, is the amount of time an activity can be delayed without an impact on the downstream activities within a schedule. There are two types of float: total float and free float. Total float is the "amount of time that a schedule activity can be delayed or extended from its early start date without delaying the project finish date or violating a schedule constraint" (PMI, 2017a). Free float is the "amount of time a schedule activity can be delayed without delaying the early start date of any successor or violating a schedule constraint" (PMI, 2017a). Since the critical path is the longest path through a project schedule, the critical path will have zero float.

### Limitations of CPM Methods

CPM activity durations and network diagrams are based on estimates to create a baseline schedule to monitor project performance against. Ideally, this should allow project managers the opportunity to control the project performance by allocating additional resources and crashing or fast-tracking activities to complete a project on time. However, in practice this is very difficult to achieve. CPM has the following known limitations (Fischer et al., 2020):

- There is an assumption that durations can be accurately estimated and forecast. Estimates are based on historical data and made by separate planning departments, not the doers—the people closest to the work (Habibi et al., 2018).
- Interdependencies are not always clear up front; they evolve as more information becomes available and work is further broken down, making it difficult to differentiate between scope creep and scope refinement. This can result in lengthy and costly contractual battles between the contractor and developer (Quah & Prabhakar, 2008).
- Schedule logic in CPM is largely driven by finish-to-start dependencies, making other types of dependencies difficult to forecast and the critical path difficult to identify without software. Software requires user training/results in a designated "scheduler," removed from the work (Ouelhadj & Petrovic, 2009).

Management assumes that projects can add resources to speed up completion, failing to account for learning curves, increased complexity, and additional bottlenecks (Brooks, 1975).

## Approach

By applying agile principles to the CPM approach, a modified approach to CPM can be used for agile software projects. However, this approach may not be appropriate for all projects; there is a continued need to have integrated schedules and milestones within government projects. Often, an agile software project only represents one piece of an overall system of systems, making it critical for agile software projects to have a method or approach to address CPM.

Agile is not a process but rather an umbrella term used to refer to several methodologies that emphasize the four agile methods and 12 principles of the Agile Manifesto developed in the early 2000s. Agile focuses on iterative and incremental development and improvements by embracing change as new information becomes available (Agile Alliance, n.d.). Traditional change management approaches are too slow and do not allow projects to respond to change rapidly.

Agile also promotes collaboration and self-organization. This means that individuals are empowered to self-organize and take ownership of their work. Decision-making is delegated to the lowest level of authority possible to enable the people doing the work to make more realistic estimates about the time and resources required to accomplish the work. To apply agile methods to project controls, CPM has to account for rolling-wave and capacity-based planning, iterative development cycles, and incremental changes.

### Production Management Concepts

All CPM schedules start with a basic list of tasks, predecessors, and durations. A network diagram to capture the overall flow of work is used to demonstrate the process from the end to end. The limitations of CPM, as listed in Limitations of CPM Methods, create the need for deeper understanding of the work that is planned and estimated to be accomplished. In order to achieve this level of detailed planning in an iterative, agile approach, production management concepts such as the production design, process design, capacity planning, inventory or work in process (WIP) control, and variability need to be taken into account during CPM planning (Project Production Institute [PPI], n.d.).

Production and process design are closely related. *Production design* is a compilation of the requirements that will be used to design and build a product (PPI, n.d.). In the case of an agile software project, the requirements are often user-based and developed in an iterative manner over time, incorporating user and stakeholder feedback throughout the design and development process. *Process design* is the design of the processes and procedures to accomplish the work within the project (PPI, n.d.). In the case of agile software projects, the agile ceremonies and cadences of sprint planning, execution, demo/review, and retrospective are examples of processes that support the development of a product. You cannot fast-track or shortcut these processes to deliver a product faster; they are necessary for achieving the requirements in a cost-effective and iterative manner. In the case of CPM, production and process design impact the overall workflow and network diagram used to build the schedule. An understanding of both the product requirements and the process requirements is needed to capture all the applicable work within a timebox, such as a sprint.

Capacity planning is the approach of developing sprint plans based on resource availability (PPI, n.d.). It requires estimation and assignment of work based on the inputs available (data, models, etc.), the number of hours available by each team member, and the complexity/scope of the work to be accomplished. Work is estimated and planned to the appropriate scope within the sprint based on the team’s capacity (Cohn, 2005). In the case of CPM, capacity drives the duration estimates, accounting for resource hours available, dependencies of the work within the network, and possible blockers (such as external reviews).

By adopting agile principles, project managers can redesign the CPM approach to account for rolling-wave planning, iterative development, and incremental changes, resulting in improved project performance.

### Rolling-Wave and Capacity-Based Planning

Rolling-wave planning is an iterative and incremental approach to project cost and schedule estimation. The estimate is done progressively over time, refined through rolling-wave planning methods that deliver cost and schedule estimates at the task level just in time. As more information becomes available and plans become more detailed, the estimate is updated and refined (Slinger, 2012). Capacity-based planning is the delegation of task estimation to the individuals conducting the work. This allows for a more detailed estimate based on the individuals’ hours available, skill sets, and personal understanding of the work complexity.

Story point analysis involves assigning story points, a unit of measurement for expressing a software development effort, a point value for the effort to complete the item. The individual values are not important. However, the relative value that the software development team assigns to the story points reflects the effort of the tasks involved. Something that is assigned a 2 should be roughly twice as much effort as something assigned a 1 (Cohn, 2005).

There are many agile estimation techniques, such as story point estimating, that allow for task estimation to account for not only the estimate hours, but also the complexity and risk associated with the task requirements (Agile Alliance, n.d.). For example, if a task is not well defined and an individual does not have the experience associated with the task, the estimate would be high, indicating to the project manager that there is an immediate need to spend more time on the task definition. Before the work even starts (resources, cost, etc.), the project manager can control the project direction by addressing the issues in real time versus finding out about it later in a monthly CPM report.

### Iterative Development Cycles and Incremental Changes

Iterative development cycles and incremental changes are similar to Plan-Do-Check-Act (PDCA) cycles. An example of an agile application of PDCA is through the scrum sprint cycles (see Figure 2). A sprint is a timeboxed period to do work, usually between 2 to 4 weeks long. A sprint starts with a planning session focused on near-term and priority work from the project backlog. Capacity-based estimates are used to determine what resources are available and how much work to plan into the sprint cycle based on the hours and skills available. The team then commits to a set amount of work for the sprint and works down the sprint backlog until the sprint ends. Even if all the work is not completed, the sprint will end on a set date. At the end of the sprint, the team reviews their work completed with their stakeholders, providing an opportunity to either continue forward or make an incremental change. This allows for rapid, responsive changes before too many resources and too much time have been committed to a flawed or problematic solution. It also provides an opportunity for stakeholders to clarify their requirements as more information becomes known, allowing for a collaborative solution to be developed (Agile Alliance, n.d.).

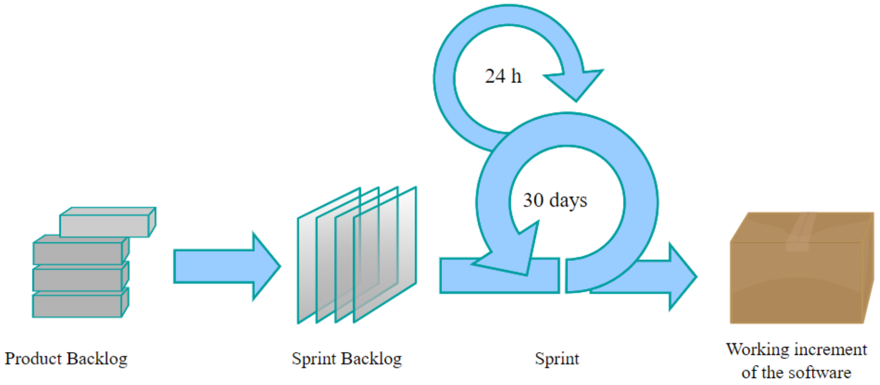

Figure 2. Scrum Sprint Cycle

### Work in Process, Inventory/Backlog, and Resource Controls

Planning work in sprints allows projects to better manage and control WIP, inventory, and resources. Large amounts of WIP leads to less throughput, as too many tasks are started but not necessarily finished. This commits resources, time, and money unnecessarily and inefficiently, which increases project variability. As project variability increases, monitoring and control methods lose their effectiveness to control, becoming historical snapshots (backward-facing), instead of forward-looking. In addition, large amounts of inventory also tie up resources and drives up project costs, as inventory requires space and resources to store, track, and manage. Finally, since agile promotes rolling-wave and capacity-based planning, work is committed based on resource availability, skill set, and capacity. Resource surges may still be needed on occasion, but they are the last resort, relying instead on the people doing the work to only commit to what they are capable of achieving within the sprint.

### Agile Approach for CPM

An agile approach for CPM also begins with rolling-wave planning. It is unlikely that most projects can produce a detailed level sequence of activities months and years in advance. However, creating the network diagram during a sprint timebox will generate less variability because resources commit to work based on their capacity and skill set. Sequencing of activities also improves because the inputs/outputs of near-term work must be well understood to estimate and commit to a sprint. If there is uncertainty related to a specific activity, then it is raised to the attention of the project manager during planning, which can then be communicated back to the stakeholders. This iterative approach to problem solving continues until the team reaches an understanding of what the requirements are and can commit to a path forward.

The critical path can be determined at the sprint and project level but is tracked differently. At the sprint level, the estimated duration can be inferred from the story point estimates for each task. The earliest start/finish and late start/finish are calculated the same way but require that tasks within a sprint have a network diagram in place as part of planning. This means identifying the predecessor and successors of each task within a sprint. It does require a modification to traditional CPM, though, in that the resources and their capacity are directly tied to a task and its story point estimate, which will affect the network diagram development and requires review as part of the sprint planning session. Further, traditional scheduling tools, like integrated master schedules, are often outdated before they are even finished. A roadmap is an agile approach for long-term planning that captures high level milestones and project targets but is constantly updated from sprint to sprint based on how the work is progressing and based on feedback from stakeholders.

## Application to Software Projects

### Software Development Tools That Support Agile CPM

There are many software tools available that can provide an agile CPM approach for software development projects. Listing of these tools is not meant as an endorsement but rather a demonstration of how agile CPM can achieve meeting the needs of both the project team and stakeholders. According to the *14th Annual State of Agile Report*, Jira remains the most popular agile project management tool across industry (Digital.ai Software Inc., 2020). Jira allows projects to plan, organize, and track work throughout the life cycle of project. Work breakdown and dependencies are captured and managed in real time within the Jira environment, allowing the team to see what is happening in a connected (versus disconnected) state (Atlassian, n.d.). This is a significant improvement over previous project management tools such as Microsoft Excel or Project.

Further, Jira plugins such as SoftwarePlant's (2021) BigPicture and ALM Works's (n.d.) Structure both allow software development projects to pull project tasking into a Gantt-style chart/network diagram view. An agile work breakdown structure provides the basis for organizing and breaking the work down. Tasks can be linked in various relationships—such as end-to-end, end-to-start, start-to-start, and start-to-end—similar to Microsoft Project or Primavera scheduling software. Milestones can be added, and work can be connected to these dates, allowing project managers to pull reports on the critical path. These data are captured and updated in real time as teams work within the system to update their tasking, eliminating the disconnect of having separate planners and doers.

### Applications Within Government Software Projects

According to the latest *Defense Acquisitions Annual Assessment*, while 22 of the DoD's weapons programs reported agile development as their software project delivery approach, there were inconsistencies across the programs in terms of execution and delivery (GAO, 2020). Out of the 22 programs, only six met the industry standard for agile delivery practices, which is delivery every 6 weeks or less (Freedberg, 2020). While this seems discouraging, it is important to note that the adoption of agile within government projects is still relatively new, and true agile transformation of a project or program can take years.

As the DoD is transitioning to more agile practices, there are multiple programs and projects that are adopting agile practices and software successfully. For example, Joint Space Operations Center (JSpOC) Mission System (JMS), Air Force; Distributed Common Ground System–Navy (DCGS-N) Increment 2, Navy; Global Combat Support System–Joint (GCSS-J), Defense Information Systems Agency (DISA); and Catapult/Attack the Network Tool Suite (ANTS), Joint Improvised-Threat Defeat Organization (JIDO) programs have all adopted Jira project management software to support iterative and incremental development practices (Kramer & Wagner, 2019). While it is unknown what, if any, Jira plugins are being used to track the critical path, the adoption and native functions within the basic version of Jira support the overall goals and objectives of CPM by allowing task tracking in an integrated and networked fashion.

## Conclusion

Agile methods allow projects to adapt to rapid and constant change. Traditional methods for project monitoring and control are based on theories and assumptions that change and variability are rare in projects, which results in project systems that can only monitor the past, not control the present or forecast the future. By applying agile methods—such as rolling-wave and capacity-based planning and iterative development cycles and incremental changes—and by focusing on controlling WIP, inventory, and resources, CPM can be improved to provide control, not just monitoring.

In an agile construct, the intent of CPM—tracking the critical path of a project—can be done within native software project management tracking tools, such as Jira. Further adoption of Jira and Jira plugins, such as BigPicture and Structure, could allow government projects to capture the detailed level schedule and critical path in real time, eliminating or greatly reducing the barriers in place today with traditional CPM approaches and tools. After all, agile and CPM share a common goal of delivering a product or project with transparency of the time/cost throughout the life cycle.

## References

Agile Alliance. (n.d.). *Agile 101*. Retrieved 2018, from https://www.agilealliance.org

Atlassian. (n.d.). *Jira software*. Retrieved April 2021, from https://www.atlassian.com/software/jira?&aceid=&adposition=&adgroup=56993361780&campaign=1439934479&creative=429724101024&device=c&keyword=jira&matchtype=e&network=g&placement=&ds_kids=p34164036185&ds_e=GOOGLE&ds_eid=700000001558501&ds_e1=GOOGLE&gclid=Cjw

ALM Works. (n.d.). *Structure: Project management at scale*. Atlassian Marketplace. Retrieved April 2021, from https://marketplace.atlassian.com/apps/34717/structure-project-management-at-scale?hosting=cloud&tab=overview

Brooks, P. F. (1975). *The mythical man-month.* Addison-Wesley Publishing Company.

Cohn, M. (2005). *Agile estimating and planning.* Pearson Education.

Digital.ai Software Inc. (2020). *14th annual state of agile report.* State of Agile. https://stateofagile.com/#ufh-i-615706098-14th-annual-state-of-agile-report/7027494

GAO. (2020, June). *Defense acquisitions annual assessment: Drive to deliver capabilities faster increases importance of program knowledge and consistent data for oversight.* https://www.gao.gov/assets/710/707359.pdf

Habibi, F., Birgani, O. T., Koppelaar, H., & Radenovic, S. (2018). Using fuzzy logic to improve the project time and cost estimation based on project evaluation. *Journal of Project Management*, 183–196.

Fischer, G., Ballard, H. G., Spearman, M., & Parsons, M. (2020, October 21). *Does CPM scheduling work?* Project Production Institute. https://projectproduction.org/events/does-cpm-scheduling-work/

Freedberg, S. J., Jr. (2020, June 12). *DoD "agile" software development still too slow: GAO*. Breaking Defense. https://breakingdefense.com/2020/06/dod-agile-software-development-still-too-slow-gao/

Kramer, J. D., & Wagner, L. J. (2019). Developmental test and requirements: Best practices of successful information systems using agile methods. *Defense Acquisition Research Journal*, *86*. https://www.dau.edu/library/arj/ARJ/ARJ88/ARJ-88%20_18-804%20Kramer.pdf

Ouelhadj, D., & Petrovic, S. (2009). Survey of dynamic scheduling in manufacturing systems. *Journal of Scheduling*, *12*(4), 417.

Project Management Institute. (2017a). *A guide to the project management body of knowledge (PMBOK Guide)* (6th ed.). Project Management Institute.

Project Management Institute. (2017b). *PMI lexicon of project management terms v3.2*. https://www.pmi.org/pmbok-guide-standards/lexicon

Project Production Institute. (n.d.). *Project Production Institute*. Retrieved September 2020, from https://projectproduction.org/

Quah, J., & Prabhakar, G. P. (2008). Scope creep in software development. *Journal of Social Management*, *6*.

Slinger, M. (2012). *Agile estimation techniques* [Presentation]. Project Management Institute Global Congress, Vancouver, Canada.

SoftwarePlant. (2021, April). *BigPicture: Project management & PPM*. Atlassian Marketplace. https://marketplace.atlassian.com/apps/1212259/bigpicture-project-management-ppm?hosting=cloud&tab=overview

Acquisition Research Program
Graduate School of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net