## Introduction to Cost and Schedule Management

In this module, we will focus on topics surrounding cost and scheduling. After reviewing cost and scheduling from a management perspective, we will continue to explore project scheduling further by looking at Rolling Wave planning. In the last section of the module, we will discuss cost and scheduling management techniques.

Two critical functions of a Software Project Manager are staying on top of the ever-fluctuating cost and schedule of a project. We will begin by looking at methods or tools used to manage the cost and schedule and we will discuss industry standards in these areas.

## Scope of Cost and Schedule Management

A Cost and Schedule Management System (CSMS) is an integrated management control structure used to manage programs that includes planning, accounting, budgeting, scheduling, work authorization, and analysis. CSMS is widely used especially on government programs but it is becoming increasingly widespread on commercial programs. Earned Value Management (EVM) is a project management method for assessing project performance and progress in an unbiased fashion. At the Department of Defense (DoD) and other Federal agencies, EVM is often referred to as the Cost/Schedule Control Systems Criteria (C/SCSC). In 1999, the United States Office of Management and Budget (OMB) began to require EVM use across all government agencies for internally managed projects and for contractors. Contractors' internal systems must meet the US government criteria to be considered acceptable by government procuring agencies.

Project Management Institute (PMI) is one of the world's largest professional membership associations. Its standards are targeted at projects, programs, people, organizations, and the profession. PMI publishes *A Guide to the Project Management Body of Knowledge* (PMBOK Guide), an industry standard, and certifies individuals as Project Management Professional (PMP).

The goal of software cost and schedule management is to ensure that quality software products are produced within the cost and schedule constraints of the program. Cost and schedule management can apply to all projects: large or small, government or commercial, risky or risk free. The amount of rigor will vary from project to project. For example, on a government program, your contract will likely require cost and schedule management, but on a commercial program, it may not be mandatory. The contract may direct specific practices to follow. If practices are not identified, the Program Software Manager must tailor practices to the program/project and obtain concurrence from the Program Manager and the Software Department Manager to ensure solid communication among the teams. The Program Software Manager is responsible for implementing the cost and schedule management program.

The benefits of cost and schedule management are improved planning, clearly defined roles and responsibilities for team members, improved problem traceability should issues arise, the ability to affect the future, early warning of potential problems, and more accurate estimates at completion (EAC).

## Management Reviews

Management reviews are conducted throughout the development effort to monitor cost and schedule progress. You should become familiar with two management reviews.

The ***Cost and Schedule Status Review*** is conducted monthly for each open Work Authorization Delegation or Document (WAD), the document that permits the tasks to be executed. The WAD Manager, the Program Manager, and the Financial Analyst must attend the meeting. The Cost Account Schedule

(CAS), which includes a detailed schedule of the current project, is examined and credit is given for task completion. A comparison is made between the work performed, work scheduled, and actual costs to determine the schedule and cost variances. This is the measure of program performance

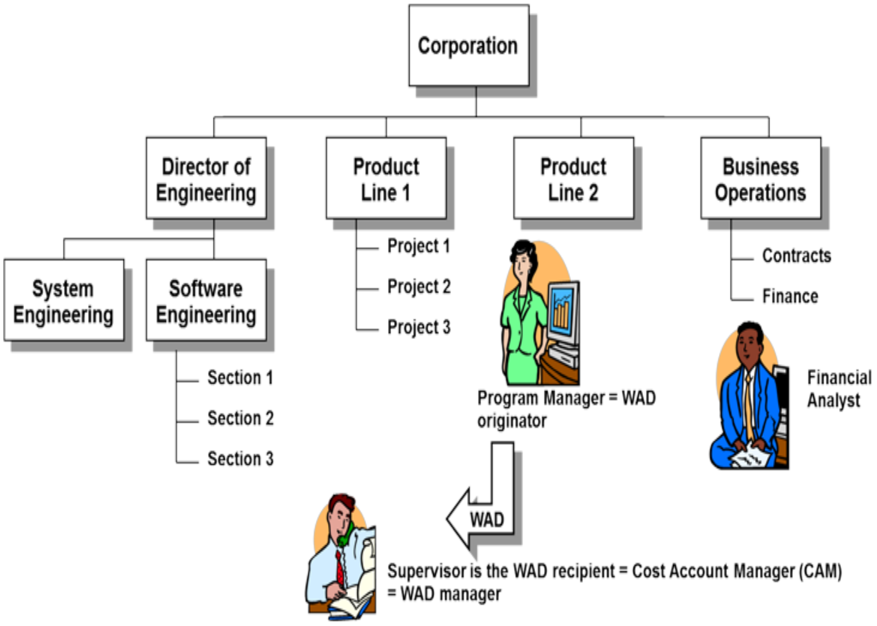
The ***WAD Review*** is also conducted monthly. Here the Engineering Manager chooses which selected programs will be evaluated for that month based on program priority, criticality, or issues. This review addresses the current schedule, problems, solutions, and plans for the future. The Engineering Director, Program Manager, Software Project Manager, and the Lead Systems/Software Engineering attend the WAD Review.

The Software Manager's responsibilities for the group's activities and at these reviews include maintaining cognizance of the status of the work accomplished and hours expended for all of his/her WADs and Work Package Tasks (WPTs), the activities listed in the WAD. This can be accomplished with weekly status reports, weekly status meetings, reviewing the financial reports (such as the Program Cost Status Report and the Program Hours Worked Report), continuously monitoring risk items, assessing issues often by using the Quality Assurance team (an independent entity), and keeping the Department Manager informed of progress at least weekly.

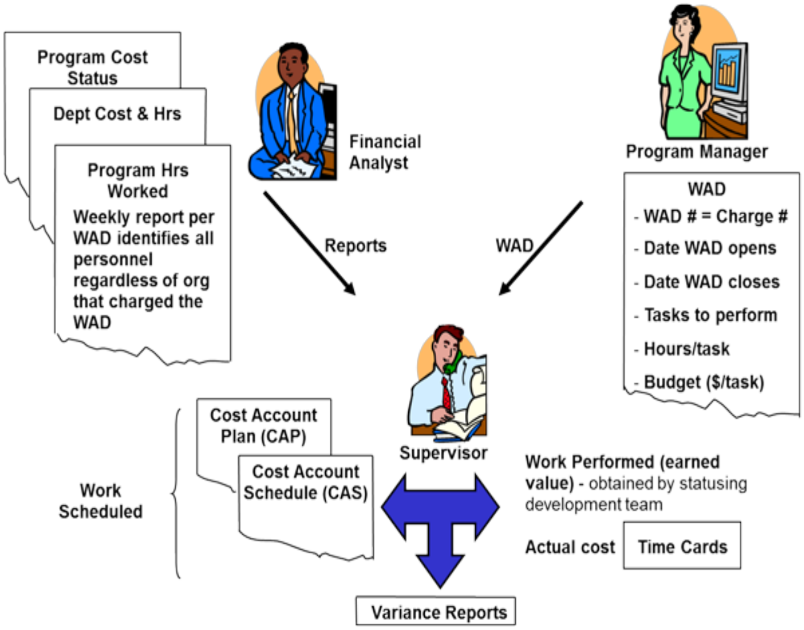
## Cost and Schedule Management Techniques

## Definitions

Within the organization, personnel have certain roles that include how they must interact with each other. The graphic below illustrates this point using CAM and others interacting with the WAD.



The image below depicts some of the many processes that must be followed to help ensure successful outcomes.



The 27 cost and schedule management terms are divided into four (4) categories as follows:

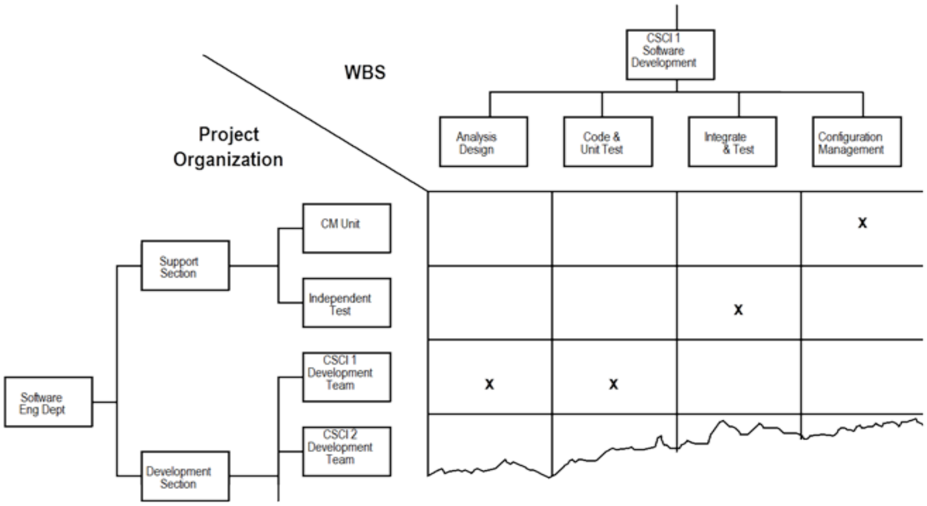
| Roles | Documentation | Process | Other |
|---|---|---|---|
| *Cost Account Manager (CAM)* | *Cost Account Plan (CAP)* | *Actual Cost* | *Direct Labor* |
| *Financial Analyst* | *Cost Account Schedule (CAS)* | *Actual Cost of Work Performed (ACWP)* | *Fiscal Year* |
| *WAD Manager* | *Department Cost Status Report* | *Budgeted Cost of Work Performed (BCWP)* | *Fringe* |
| *WAD Originator* | | *Budgeted Cost of Work Scheduled (BCWS)* | *General and Administrative (G&A)* |
| *WAD Recipient* | *Funds Expenditure Report* | *Cost Variance* | *Indirect Labor* |
| | *Program Cost Status Report* | *Earned Value* | *Overhead* |
| | *Program Hours Status Report* | *Rolling Wave* | |
| | *Variance Report* | *Schedule Variance* | |
| | *Work Authorization Delegation or* | *Work Accomplished* | |

| Roles | Documentation | Process | Other |
|---|---|---|---|
| | *Document (WAD)* | | |
| | *Work Breakdown Structure (WBS)* | *Work Allocated* | |
| | *Work Package Task (WPT)* | *Work Budgeted* | |
| | | *Work Performed* | |
| | | *Work Planned* | |
| | | *Work Scheduled* | |

*See Definitions at the bottom of Course Modules for definitions of these terms.*

The Program Manager is the WAD originator and the supervisor is called the WAD recipient, the Cost Account Manager (CAM), or the WAD manager. Determining the status of the development team's work gives you the Earned Value or Work Performed.

Programs create a Work Breakdown Structure (WBS) or discrete work elements organized in a way that helps define the total work scope of the project as shown in the graphic below.

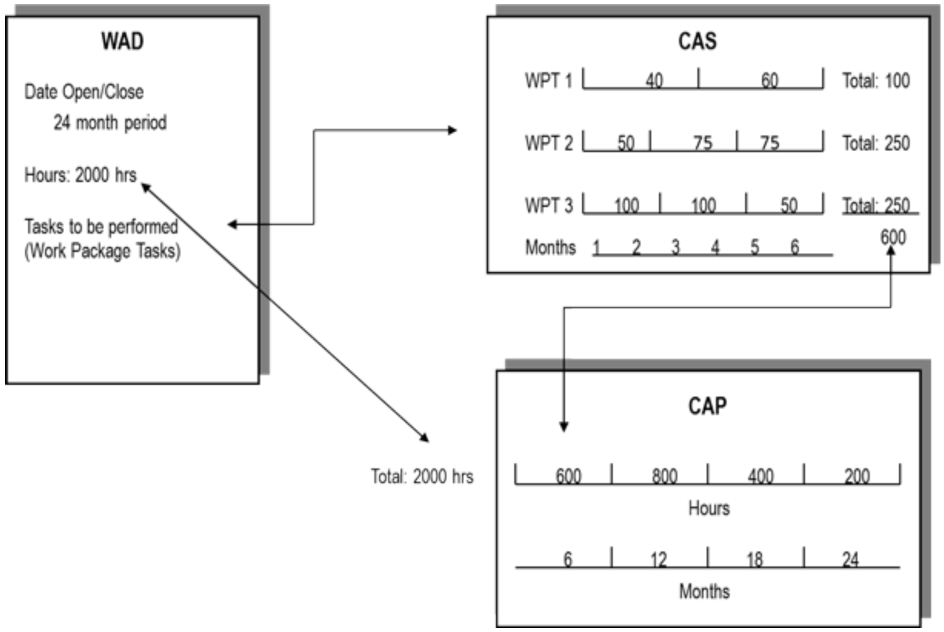


When the Software Manager receives a WAD, he/she assigns the WBS elements or Work Package Tasks (WPTs) to individuals to perform.

## Aligning the WAD, CAP, and CAS

The WAD contains the tasks—generally—broken out by each phase of the project lifecycle. The Cost Account Schedule (CAS) is used to schedule milestones for each WPT over a near term period of time (typically 6 months). Each WPT is allocated hours in the schedule for each month it is performed.

The Cost Account Plan (CAP) is used to show how the contract hours (not yet allocated to CAS) are allocated over the remaining months of the schedule. The CAS & CAP are used together to monitor the amount of work performed (earned value) against work scheduled for each Work Package Task (WPT) defined in the CAS. The WPTs defined in the CAS must match the task description in the WAD. The definition of WPTs is critical to the software cost & schedule management process. The following graphic shows how hours on the WAD reflect on the CAP and CAS when applied.



## Earned Value Management (EVM)

A cost and schedule management implementation is used to analyze.

**schedule variance** = earned value (or work performed) less work scheduled

and

**cost variance** = earned value (or work performed) less actual cost.

CSMS requires the use of Cost Account Plans (CAPs) and Cost Account Schedules (CASs). It is typically applied to development programs that have less than 10% level-of-effort (LOE) tasking.

LOE is defined as support type activities such as management or clerical activities with no definite or deliverable products. The budget is evenly distributed over period of performance so the activity is never ahead of or behind schedule—that is, the earned value is based on passage of time.

To monitor cost and schedule, formal meetings are held periodically (generally monthly) with the financial analyst, CAM, and Program Manager to compare the amount of work performed (claimed by the CAM) with the amount scheduled (identified in the CAS and CAP) and actual cost of work performed (obtained from employee time cards).

### Five Earned Value Methods

Earned Value is the same as Accomplishment or Work Performed.

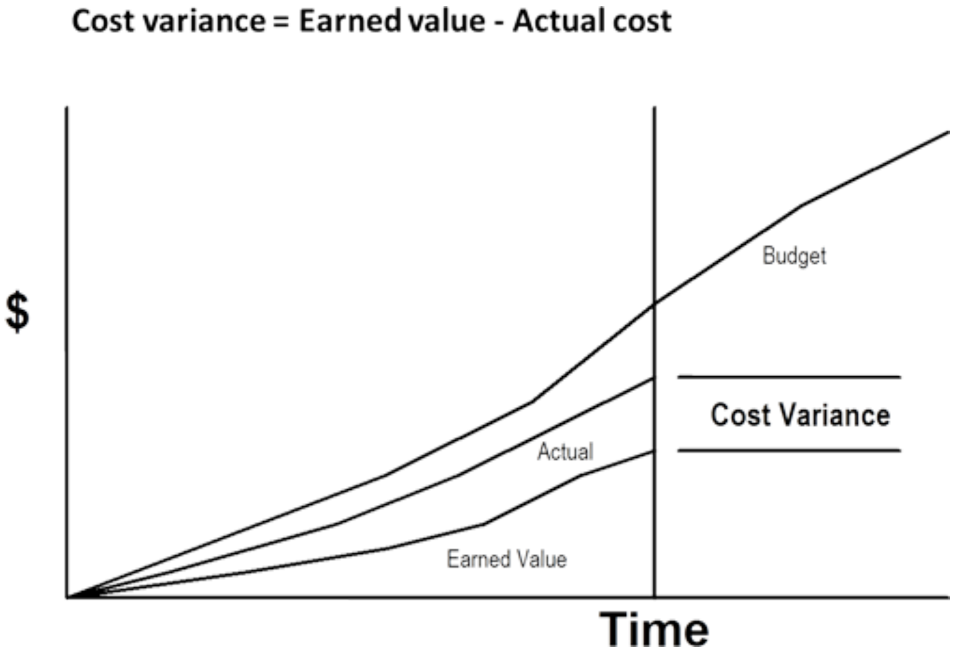
The CAM selects the earned value method and budgets the work to be done for each allocated WAD WPT. There are five widely used earned value methods.

The next item contains a video that provides a brief overview of each of the five most common methods.

## Cost and Schedule Variance

As the project progresses, CAMs monitor progress by comparing earned value (or work performed) with the plan (or budget from CAS) and with actual cost (time cards & accounting system) to determine schedule and cost variance. This is done monthly for the preceding month and cumulatively for the schedule to date. Significant variances (top 10 variances or those exceeding some percentage, often 10%) are analyzed and documented in a cost or schedule variance report. The variance report includes what caused the problem, the corrective action planning, and the impact on the project.

Cost Variance is defined as the earned value minus the actual costs as illustrated below.



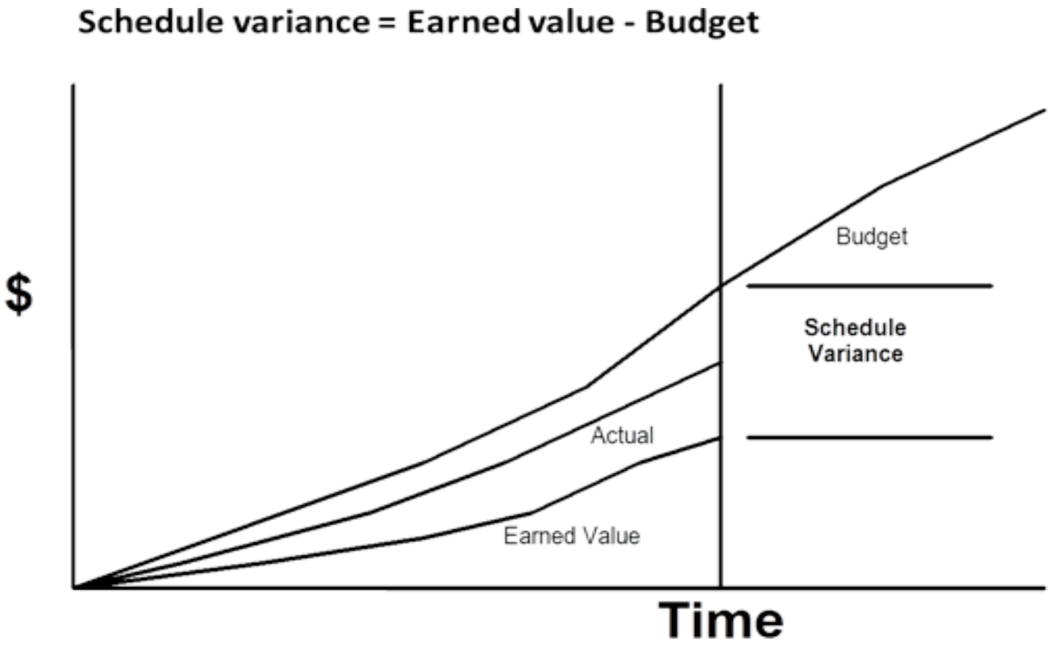
Actual cost consists of direct labor, fringe benefits, overhead, and general and administrative (G&A) costs as shown here.

| Actual Cost of the Product is: | Example |
|---|---|
| Direct Labor | $30.00/hr |
| Fringe (benefits) 40% | $12.00/hr |
| Labor and Fringe Subtotal | $42.00/hr |
| Overhead (office space, utilities, etc.) 50% | $21.00/hr |
| Labor, fringe, and overhead Subtotal | $63.00/hr |
| General & Administrative costs (G&A) 10% | $6.30/hr |

| Actual Cost of the Product is: | Example |
|---|---|
| Overall Total Billing Rate | $69.30/hr |

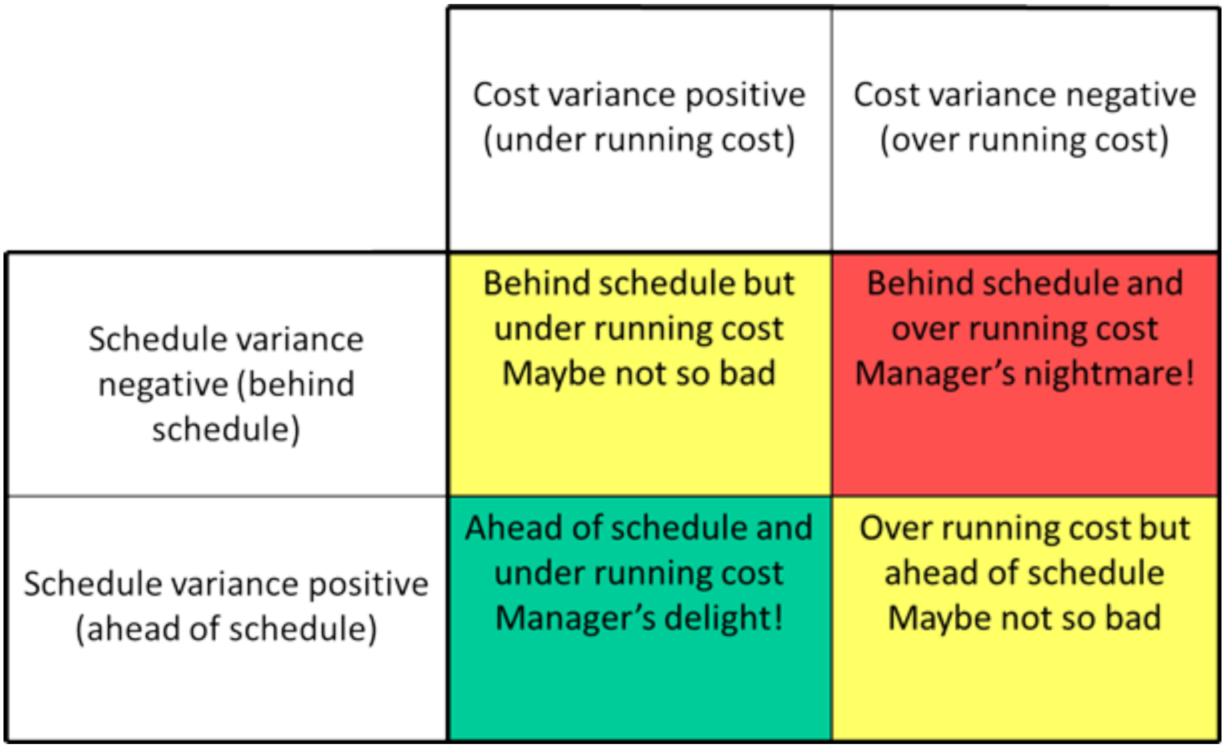
Projects may allow different combinations of direct labor, fringe benefits, overhead, and/or G&A costs to be charged to the customers and companies will vary in the percentage charged for fringe benefits, overhead and G&A costs.

Schedule Variance is defined as the earned value minus the budget, as shown here.



## Effect of Cost and Schedule Combinations

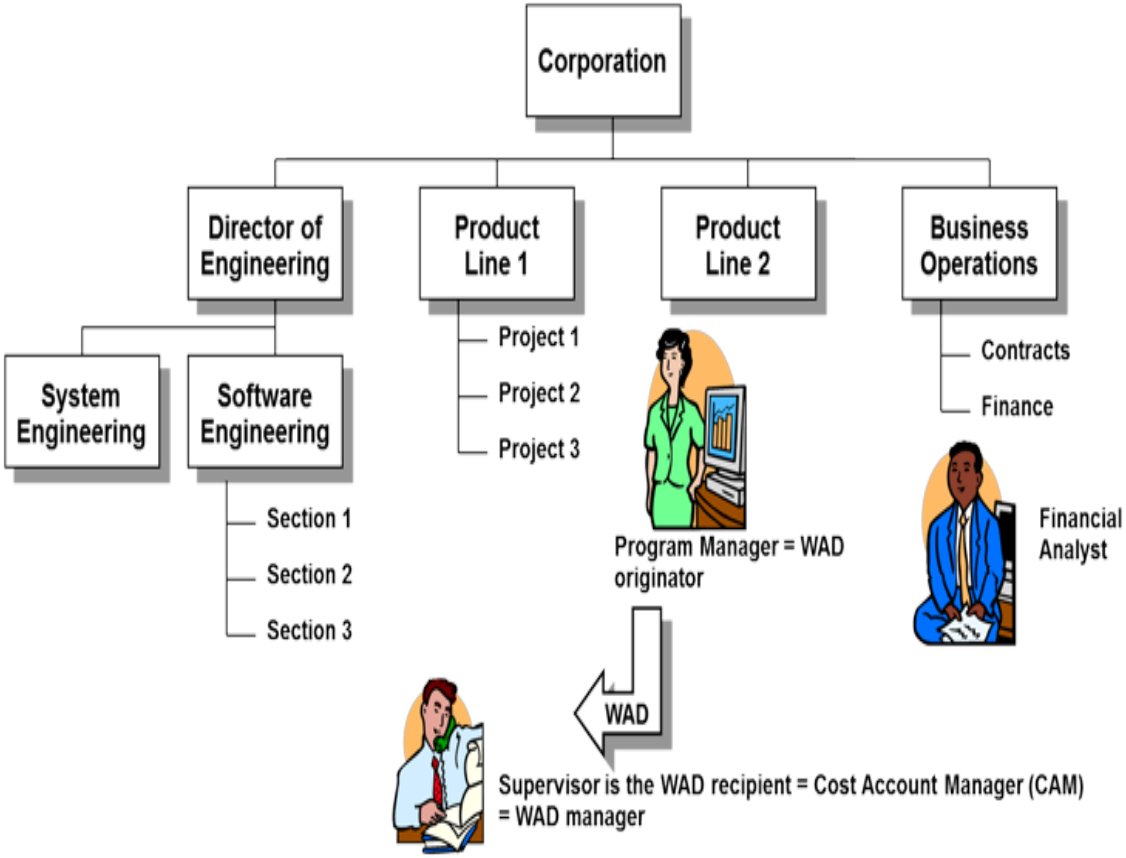
The purpose of these tools, methods, and standards is to help you, as the project manager, keep account of your project's status. Being ahead or behind in cost or schedule impacts the project and can have a ripple effect on future tasks and the entire project. The chart below shows the various combinations.

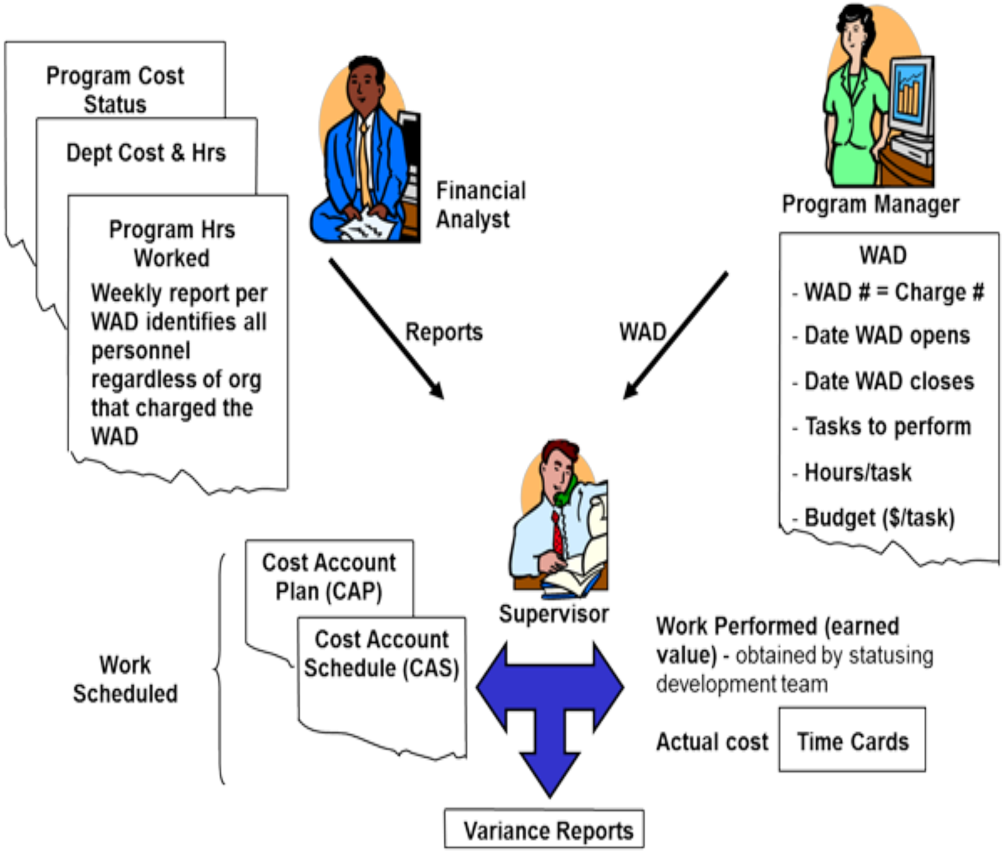


| | Cost variance positive (under running cost) | Cost variance negative (over running cost) |
|---|---|---|
| Schedule variance negative (behind schedule) | Behind schedule but under running cost<br>Maybe not so bad | Behind schedule and over running cost<br>Manager's nightmare! |
| Schedule variance positive (ahead of schedule) | Ahead of schedule and under running cost<br>Manager's delight! | Over running cost but ahead of schedule<br>Maybe not so bad |

# Earned Value Management

# Earned Value Management Definitions





Program Cost Status

Dept Cost & Hrs

Program Hrs Worked

Weekly report per WAD identifies all personnel regardless of org that charged the WAD

Financial Analyst

Program Manager

WAD
- WAD # = Charge #
- Date WAD opens
- Date WAD closes
- Tasks to perform
- Hours/task
- Budget ($/task)

Reports

WAD

Supervisor

Work Scheduled

Cost Account Plan (CAP)

Cost Account Schedule (CAS)

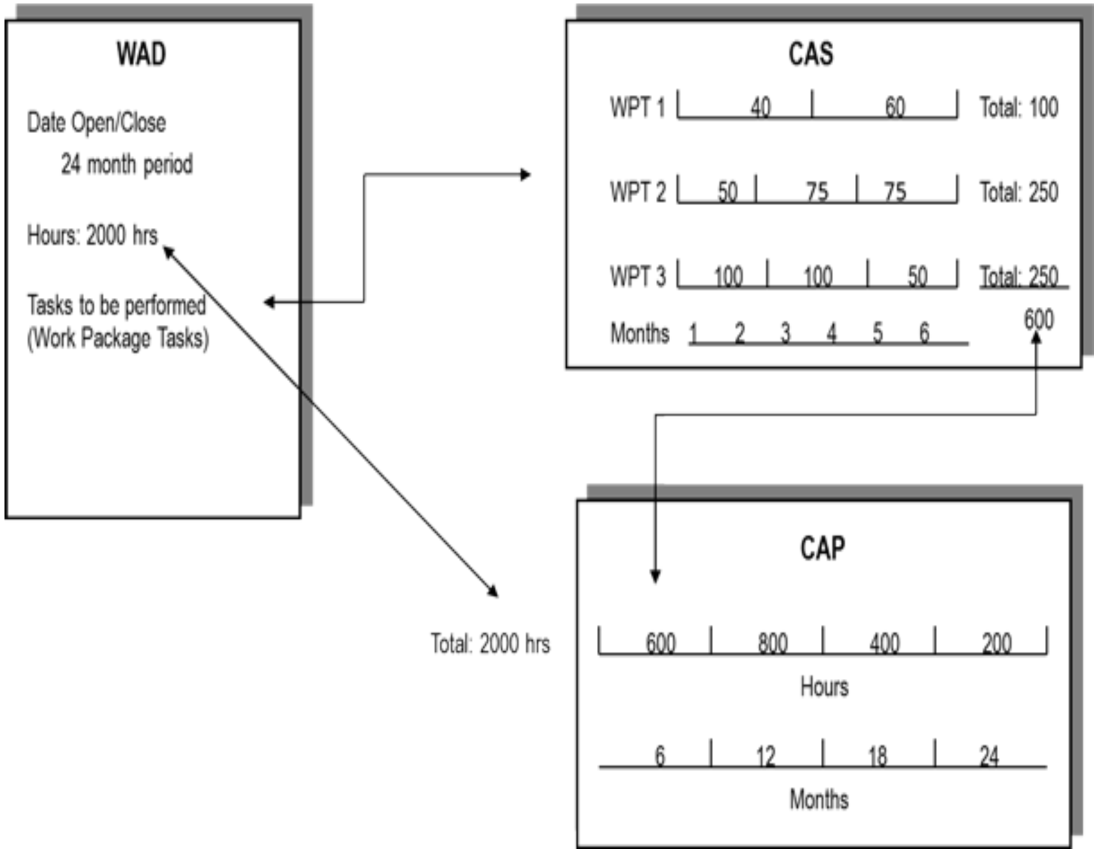
Work Performed (earned value) - obtained by statusing development team

Actual cost

Time Cards

Variance Reports

# Cost Account Schedule and Cost Account Plan



WAD

Date Open/Close
24 month period

Hours: 2000 hrs

Tasks to be performed
(Work Package Tasks)

CAS

WPT 1 40 60 Total: 100

WPT 2 50 75 75 Total: 250

WPT 3 100 100 50 Total: 250

Months 1 2 3 4 5 6

600

CAP

Total: 2000 hrs

600 800 400 200

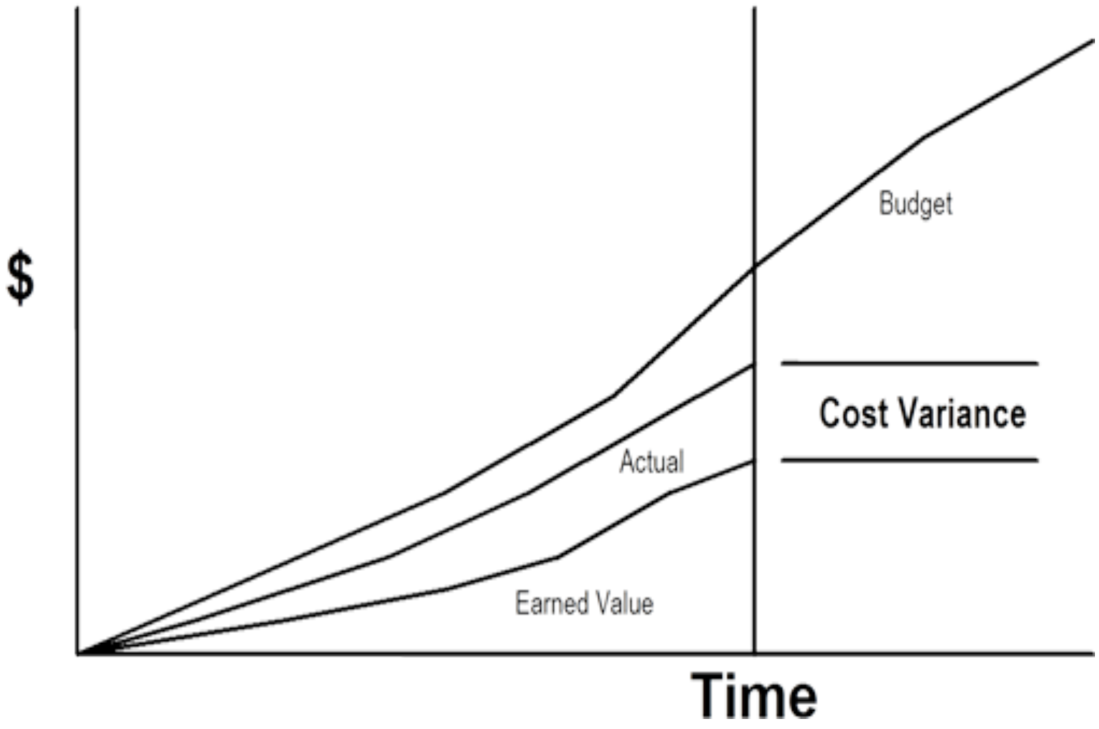
Hours

6 12 18 24

Months

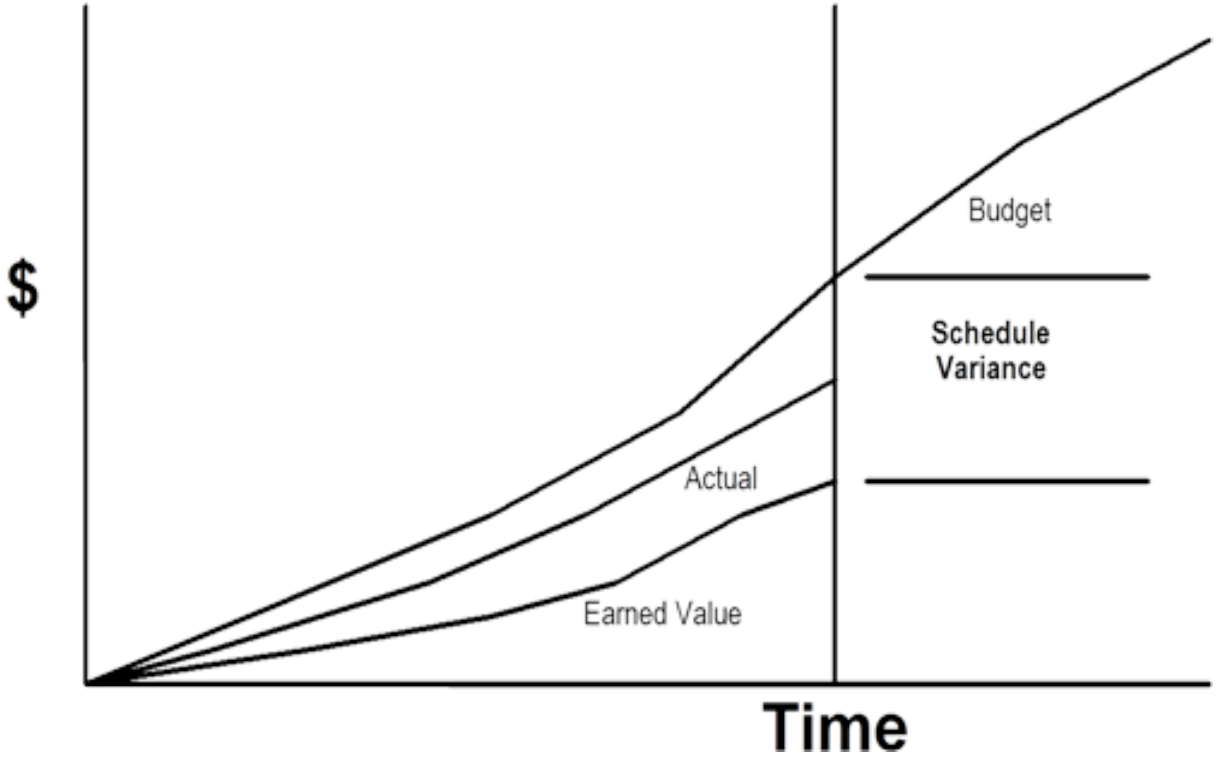
# Cost Variance

Cost variance = Earned value - Actual cost

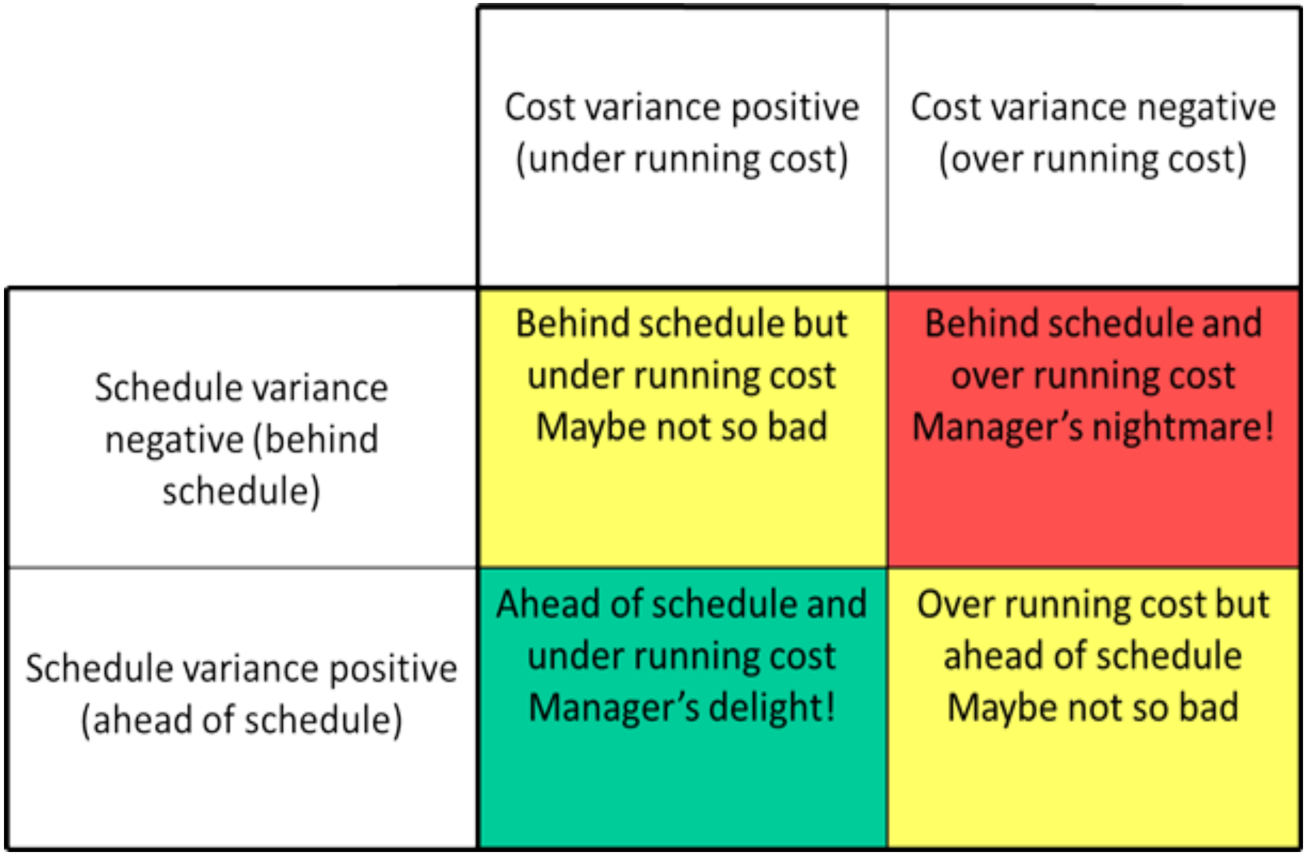


# Schedule Variance

Schedule variance = Earned value - Budget



# Effect of Cost and Schedule Combinations

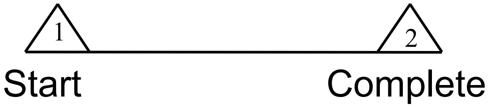


| | Cost variance positive (under running cost) | Cost variance negative (over running cost) |
|---|---|---|
| Schedule variance negative (behind schedule) | Behind schedule but under running cost<br>Maybe not so bad | Behind schedule and over running cost<br>Manager's nightmare! |
| Schedule variance positive (ahead of schedule) | Ahead of schedule and under running cost<br>Manager's delight! | Over running cost but ahead of schedule<br>Maybe not so bad |

# Earned Value Management

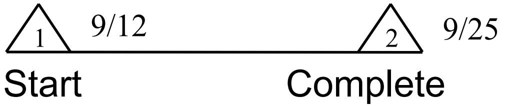
# 0/100

## Planning



| Milestone | Budget |
|---|---|
| 1 | 0 |
| 2 | 500 |
| Total | 500 |

## Status



| Milestone | Budget | Earned Value |
|---|---|---|
| 1 | 0 | 0 |
| 2 | 500 | 500 |
| Total | 500 | 500 |

- No Budget allocated to start milestone
- Total budget allocated to completion milestone
- All work claimed only when task is complete
- Spans one accounting month

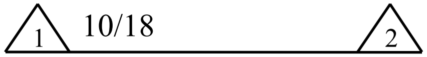
# Percent Start / Percent Complete

**Planning (50%/50%)**

1
2
Start
Complete

| Milestone | Budget |
|---|---|
| 1 | 500 |
| 2 | 500 |
| Total | 1000 |

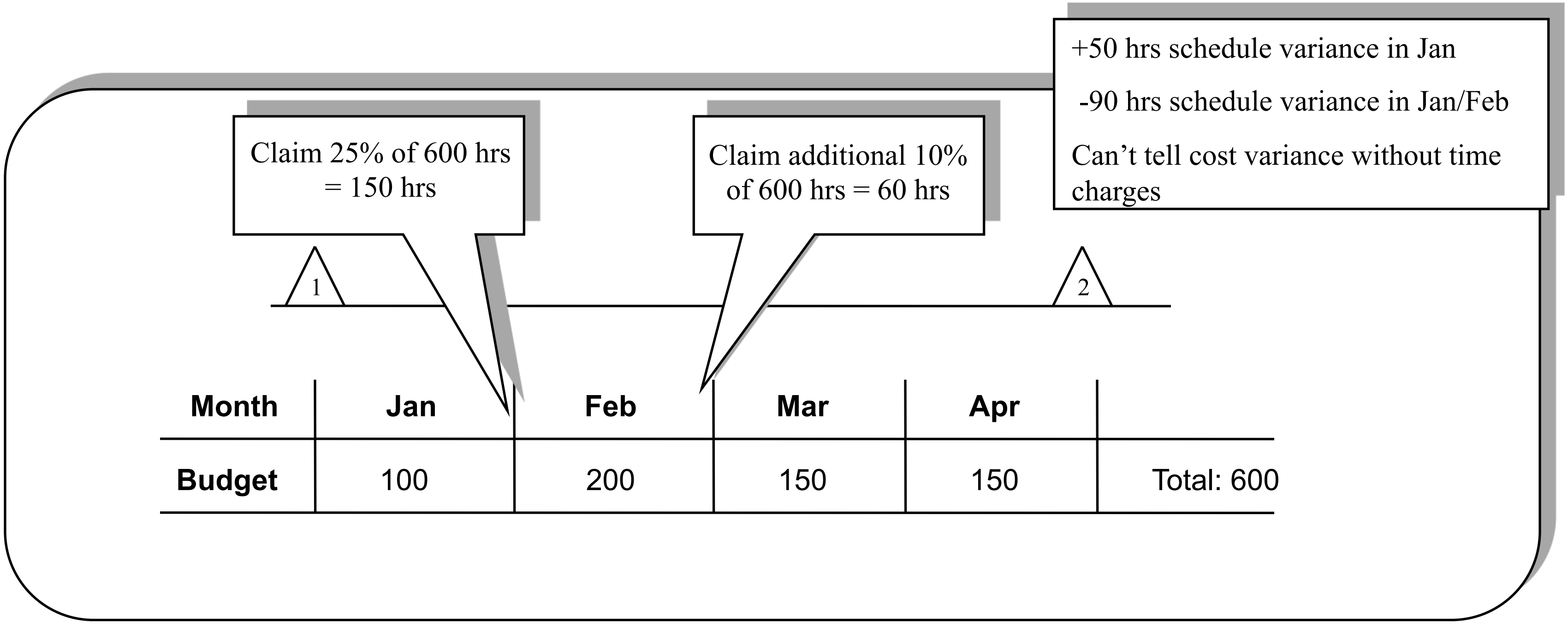
**Status**



| Milestone | Budget | Earned Value |
|---|---|---|
| 1 | 500 | 500 |
| 2 | 500 | 0 |
| Total | 1000 | 500 |

- Examples of weights:  25/75, 30/70, 50/50
- Budget is allocated to start milestone
- Budget is allocated to completion milestone
- Spans two accounting months

# Percent Complete



Claim 25% of 600 hrs = 150 hrs

Claim additional 10% of 600 hrs = 60 hrs

+50 hrs schedule variance in Jan

-90 hrs schedule variance in Jan/Feb

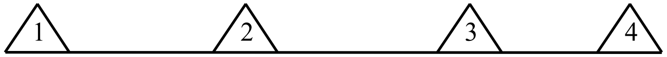
Can’t tell cost variance without time charges

| Month | Jan | Feb | Mar | Apr | |
|---|---|---|---|---|---|
| Budget | 100 | 200 | 150 | 150 | Total: 600 |

- Status of work depends on manager’s assessment of amount complete
- End of January, CAM claims work is 25% done (150 hours)
- End of February, CAM claims only another 10% (60 hours)

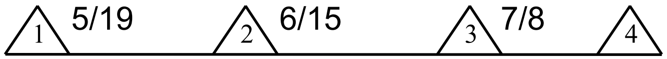
# Milestone Weights

## Planning



| Milestone | Budget |
|---|---|
| 1 | 500 |
| 2 | 500 |
| 3 | 700 |
| 4 | 300 |
| Total | 2000 |

## Status



| Milestone | Budget | Earned Value |
|---|---|---|
| 1 | 500 | 500 |
| 2 | 500 | 500 |
| 3 | 700 | 700 |
| 4 | 300 | |
| Total | 2000 | 1700 |

- Weight associated with each milestone is earned when milestone is complete
- At least on milestone per month

# Milestone Weights with Percent Complete

- Similar to milestone weights method
- Added feature allows CAM to claim percent complete activities in months without milestones
- Milestones are required at least every two months

# Characteristics of Good Milestones

- Product or event
- Must be within the authority of the CAM
- Clear, objective criteria for measuring accomplishment
- Quantified whenever possible
- Directly related to the work package (Work Authorization Delegation (WAD) and Statement of Work (SOW))

# Level of Effort

- Support activities e.g., management, clerical
- No definite or deliverable products
- Budget is evenly distributed over the period of performance
- Earned value is based on the passage of time, thus never ahead or behind schedule

## Milestones by Work Package and Phase

For the nine (9) phases defined, here are the 42 recommended work package tasks and 146 recommended milestones for each phase.

| Phase | Work Package Task | Milestones |
|---|---|---|
| 1 - System Requirements Analysis | 1 - SW engineering support for the SSS | 1 – SSS Internal draft |
| | | 2 – Revision for preliminary SSS |
| | 2 - SRR preparation & participation | 3 – Prepare draft SRR data package |
| | | 4 – Internal review of draft SRR data package |
| | | 5 - SRR data pkg revised for CDRL submission |
| | | 6 – Conduct SRR |
| | | 7 – Respond to SRR action items |
| 2 – System Design | 3 - SW engineering support for the final SSS | 8 – Start SSS updates |
| | | 9 – Complete SSS updates |
| | 4 - SW engineering support for the SSDD | 10 – Prepare draft SSDD input |
| | | 11 – Internal review of draft SSDD input |
| | | 12 – SSDD revised for CDRL submission |
| | 5 - Software Architecture Definition | 13 – Produce System Context Diagram |
| | | 14 – Define CSCI external interfaces |
| | | 15 – Generate external events list |
| | | 16 – State/mode allocation to CSCI capabilities |
| | | 17 – Determine CSCI capability IF |
| | | 18 – Establish CSCI SW development file |
| | 6 - Preliminary SW requirements spec | 19 – Prepare preliminary SRS input |
| | | 20 – Internal review of SRS input |
| | | 21 – Completion of preliminary SRS input |

| Phase | Work Package Task | Milestones |
|---|---|---|
| | 7 - Preliminary SRS Production/delivery | 22 – Integrate SRS inputs |
| | | 23 – Internal review of preliminary SRS |
| | | 24 – Publish preliminary SRS |
| | | 25 – Prepare preliminary IRS inputs |
| | | 26 – Internal review of preliminary IRS |
| | | 27 – Complete preliminary IRS input |
| | 8 - Preliminary IRS Production & Delivery | 28 – Integrate preliminary IRS inputs |
| | | 29 – Internal review of preliminary IRS |
| | | 30 – Publish Preliminary IRS |
| | 9 - Software Development Plan | 31 – Prepare SDP |
| | | 32 – Internal review of SDP |
| | | 33 – Publication of the SDP |
| | 10 - System Design Review | 34 – Prepare draft SDR material |
| | | 35 – Internal review of draft SDR material |
| | | 36 – Completion of SDR material |
| | | 37 – Publish SDR package |
| | | 38 – Conduct SDR |
| | | 39 – Work off SDR action items |
| 3 - Software Rqmts Analysis | 11 - Establish CSCI domains | 40 – Create CSCI context diagram |
| | | 41 – Generate CSCI external events list |
| | | 42 – Generate CSCI state transition diagram |
| | 12 - Conduct Software rqmts analysis | 43 – Produce complete CSCI logical model |
| | | 44 – Complete requirements traceability matrix |
| | 13 - Generate database model | 45 – Produce complete database model |

| Phase | Work Package Task | Milestones |
|---|---|---|
| | | 46 – Complete requirements traceability |
| | 14 - Finalize Software requirements | 47 – Generate data dictionary |
| | | 48 – Produce complete transformation spec |
| | 15 - Final SRS production & delivery | 49 – Integrate SRS inputs |
| | | 50 – Internal review of final SRS |
| | | 51 – Publish final SRS |
| | | 52 – Integrate final IRS inputs |
| | | 53 – Internal review of final IRS |
| | | 54 – Publish final IRS |
| | 16 - SSR preparation & conduct | 55 – Prepare SSR materials |
| | | 56 – Internal review of SSR materials |
| | | 57 – Conduct the SSR |
| 4 – Software Preliminary Design | 17 - Define Computer Software Components | 58 – Determine CSCI/CSC hierarchy |
| | | 59 – Produce CSC-level structure charts |
| | | 60 – Internal review of CSC structure charts |
| | | 61 – Establish CSC SW Development Files |
| | 18 - External interface definition | 62 – Identify message types |
| | | 63 – Define data fields that compose each msg |
| | | 64 – Internal review of message types |
| | | 65 – Complete message type definitions |
| | 19 - Human machine interface definition | 66 – Identify screens & displays |
| | | 67 – Internal review of screens & displays |
| | | 68 – Complete screen & display |

| Phase | Work Package Task | Milestones |
|---|---|---|
| | | definitions |
| | 20 - Preliminary Software Design Document (SDD) production & delivery | 69 – Integrate design inputs into prelim SDD |
| | | 70 – Internal review of preliminary SDD |
| | | 71 – Publish preliminary SDD |
| | 21 - Preliminary IF Design Document (IDD) Production & delivery | 72 – Integrate design inputs into prelim IDD |
| | | 73 – Internal review of preliminary IDD |
| | | 74 – Publish preliminary IDD |
| | 22 - Draft Software Test Plan | 75 – Produce draft STP |
| | | 76 – Internal review of draft STP |
| | 23 - Establish CSC I&T Test requirements | 77 – Start CSC I&T test requirements |
| | | 78 – Complete CSC I&T test requirements |
| | 24 - PDR preparation & conduct | 79 – Prepare PDR material |
| | | 80 – Internal review of PDR material |
| | | 81 – Conduct PDR dry run |
| | | 82 – Conduct PDR |
| | | 83 – Respond to PDR action items |
| 5 – Software Detailed Design | 25 - Complete design | 84 – Complete CSU level structure charts |
| | | 85 – Internal review of CSU structure |
| | | 86 – Complete PDL for each CSU |
| | | 87 – Internal review of CSU design |
| | | 88 – Establish CSU SW Development Files |
| | 26 - Final SDD production & delivery | 89 – Start integration of SDD inputs |
| | | 90 – Complete integration of SDD input |

| Phase | Work Package Task | Milestones |
|---|---|---|
| | | 91 – Internal review of SDD |
| | | 92 – Publication of SDD |
| | 27 - Final IDD production & delivery | 93 – Start integration of final IDD inputs |
| | | 94 – Complete integration of final IDD inputs |
| | | 95 – Internal review of IDD |
| | | 96 – Publication of final IDD |
| | 28 – Software test documentation | 97 – Integrate final inputs for the STP |
| | | 98 – Internal review of final STP |
| | | 99 – Publication of final STP |
| | | 100 – Produce draft STDs |
| | | 101 – Internal review of draft STDs |
| | 29 – CSC I&T planning | 102 – Establish CSC test responsibilities |
| | | 103 – Produce CSC test cases |
| | | 104 – Produce CSC test schedules |
| | 30 – CSU test planning | 105 – Establish CSU test responsibilities |
| | | 106 – Produce CSU test cases |
| | | 107 – Produce CSU test schedules |
| | 31 – CDR preparation & conduct | 108 – Prepare CDR material |
| | | 109 – Internal review of CDR material |
| | | 110 – Conduct CDR dry run |
| | | 111 – Conduct CDR |
| | | 112 – Respond to CDR action items |
| 6 – Coding & CSU testing | 32 – CSU code & CSU testing | 113 – Code CSUs |
| | | 114 – Produce Unit test plan for CSU |
| | | 115 – Produce code inspection |

| Phase | Work Package Task | Milestones |
|---|---|---|
| | | package |
| | | 116 – Conduct code inspection |
| | | 117 – Perform CSU testing |
| | | 118 – Produce Unit test inspection package |
| | | 119 – Conduct Unit test inspection |
| | | 120 – Submit CSU material for config mgmt |
| | 33 – CSCI integration plan | 121 – Start CSCI integration plan |
| | | 122 – Complete CSCI integration plan |
| | 34 – CSC test procedures | 123 – Start CSC I&T procedures |
| | | 124 – Complete CSC I&T procedures |
| 7 – CSC I&T | 35 – Functional string testing | 125 – Build string executable |
| | | 126 – Start string testing |
| | | 127 – Complete string testing |
| | | 128 – Resolve problems & retest as required |
| | | 129 – Document string test results for the TRR |
| | 36 – SW Test descriptions/procedures | 130 – Start SW test descriptions & procedures |
| | | 131 – Complete test descriptions & procedures |
| | 37 – Test readiness review preparation and conduct | 132 – Integrate string test data into TRR pkg |
| | | 133 – Conduct the TRR |
| | | 134 – Respond to TRR action items |
| | | 135 – Enter CSCI code into product baseline |
| 8 – CSCI Testing | 38 – CSCI testing support | 136 – None (LOE task) |
| | 39 – FCA/PCA support | 137 – None (LOE task) |
| | 40 – Product baseline documentation | 138 – Update/produce documents (SDD, IDD, SPS, VDD) |
| | | 139 – Internal review of documents |

| Phase | Work Package Task | Milestones |
|---|---|---|
| | | 140 – Correct per review and submit to CM |
| | 41 – O&M Documentation Support | 141 – Start documentation (User's Manual, Maintenance Manual) |
| | | 142 – Complete draft |
| | | 143 – Internal review of draft |
| | | 144 – Update draft based on comment review |
| | | 145 – Produce documents and submit to CM |
| 9 – System I&T | 42 – System I&T | 146 – None (LOE task) |

## Rolling Wave Planning

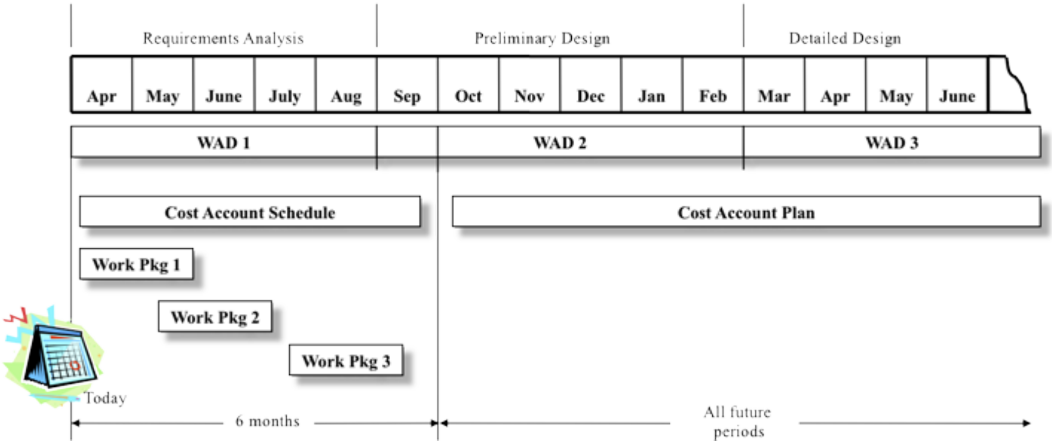
Rolling Wave Planning is a method of carefully budgeting to ensure the success of cost and schedule management. Long-term activities are difficult to plan since there are many variables. Thus, we plan in rolling waves or "just in time planning." Project plans are initially defined at a high level. As development progresses, detailed planning for each phase is performed. The detailed planning for the next phase of the life cycle is completed just before the start of that phase aptly named "just in time planning." This ensures maximum accuracy in budget planning.

### Methods of Applying Rolling Wave Planning for Software Engineering Projects

Now that WAD and Rolling Wave have been introduced to you, we will begin exploring methods of applying them.

### Establish the WADs for Each Phase in the Lifecycle

This approach takes advantage of the natural phase subdivisions and corresponding intermediate milestones of the software project and provides the best, objective work in progress measure. The image below depicts the WAD Lifecycle.



Each WAD defines the tasks to be performed and the budget allocation. If the WAD covers a period greater than some designated period, usually six (6) months, a Cost Account Plan (CAP), or high-level schedule is defined. The CAP allocates the hours for the tasks defined in the WAD and shows how contract hours are allocated over the entire time covered by the WAD. The CAS is a detailed schedule of the work to occur within the next few (usually six) months.

The WAD originator (Program Manager), Financial Analyst, and WAD manager, typically the software supervisor, agree to and sign the WAD.

### Perform the CAP to Cost Account Schedule (CAS) Conversion

Budget allotments allocated in the CAP are distributed among detailed work packages defined in the CAS (the detailed schedule) for the upcoming phase.

Budgets from multiple planning packages may be combined and redistributed and the work packages may be re-planned. This activity occurs as close to the next software phase as possible, but not later than 30 days before, and takes into account the most current program information. To achieve a sufficient level of accuracy, the planning cycle for each phase must not begin too early.

A change to the total budget allocated to the planning packages and work packages require a formal WAD revision.

The WAD Manager performs this formal process with Program Management and the Financial Analyst's oversight.

## Develop Detailed Plans for Each Phase

The traditional software development process model provides the basis for consistent phase and milestone definitions required to define the work packages for each CAS.

Suggested Lifecycle Phases

- System Requirements Analysis
- System Design
- Software Requirements Analysis
- Preliminary Design
- Detailed Design
- Code and Computer Software Unit (CSU) Testing
- CSU Integration and Testing (I&T)
- Computer Software Configuration Item (CSCI) Testing
- System Integration and Testing

You may combine phases as required by your customer or tailor as appropriate.

## Define the Progress Points or Milestones by Phase

Each software development phase has well-defined activities and measurable progress points that can be used to monitor, measure, and document progress.

Refer to the Milestones by Phase document for descriptions of the nine (9) phases, 42 recommended work package tasks, and 146 recommended milestones for each phase.

Milestones by Work Package and Phase (PDF attached to the same item as this PDF in Blackboard).

## Rolling Wave Planning Example

Suppose the Program Manager / System Engineering Manager funds the Software Engineering Organization to support System Requirements Analysis phase. This activity is scheduled for four (4) months, September 1 through December 31. The software organization is required to provide two senior software engineers (one of which is the supervisor) at the start of the task and jump to four total software engineers at the start of the second month.

The available hours/month on the task is specified as 150 hours/month. This figure is derived as follows:

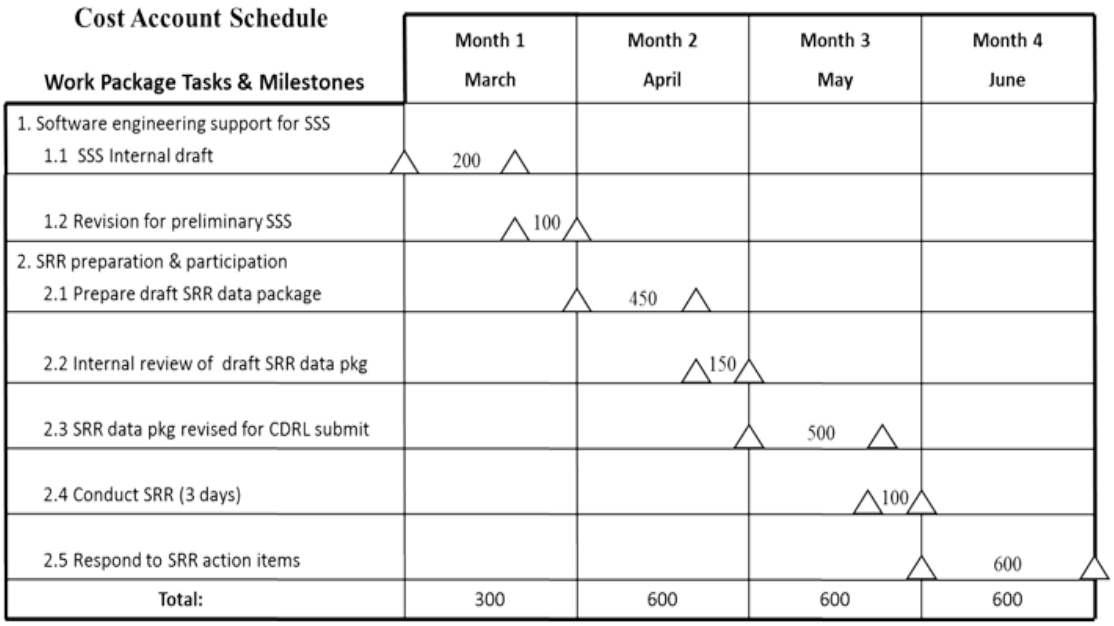
| | |
|---|---|
| Hours in the year (52 weeks x 40 hours/week) | 2080 |
| Less average Vacation hours / person (100 hours/year) 1980 | 1980 |
| Less average Holiday hours / person (80 hours/year) 1900 | 1900 |
| Less average Sick leave hours / person (100 hours/year) 1800 | 1800 |
| 1800 hours = 12 months (m) x 150 hours/month (h/m) | |

The software manager has a WAD which contains 2100 hours and can be defined as follows:

2100 h = [2 senior software engineers x (4 m x 150 h/m)] + [2 more software engineers x (3 m x 150 h/m)]

Or 2100 h = [1 m x (2 senior software engineers x 150 h/m)] + [3 m x (4 software engineers x 150 h/m)]

With this information, the software supervisor develops the following CAS:



| Cost Account Schedule<br>Work Package Tasks & Milestones | Month 1<br>March | Month 2<br>April | Month 3<br>May | Month 4<br>June |
|---|---|---|---|---|
| 1. Software engineering support for SSS<br>1.1 SSS Internal draft | △ 200 △ | | | |
| 1.2 Revision for preliminary SSS | △ 100 △ | | | |
| 2. SRR preparation & participation<br>2.1 Prepare draft SRR data package | | △ 450 △ | | |
| 2.2 Internal review of draft SRR data pkg | | △150△ | | |
| 2.3 SRR data pkg revised for CDRL submit | | | △ 500 △ | |
| 2.4 Conduct SRR (3 days) | | | △100△ | |
| 2.5 Respond to SRR action items | | | | △ 600 △ |
| Total: | 300 | 600 | 600 | 600 |