Earned Value Work Plan

CxOne User Guide

CxGuide_EarnedValueWorkPlan.doc

December 9, 2002

Construx
SOFTWARE

cxone

Advancing the Art and
Science of Commercial
Software Engineering

CONTENTS

Copyright Notice

For further information or support visit www.construx.com.

1 Introduction

This guide discusses how to use ***CxTemp_EarnedValueWorkPlan*** along with supporting practices and ideas to create and manage a *work plan* using earned value techniques.

Useful concepts and techniques for project management are described that can be integrated into existing methodologies, even if your are not using the CxOne work plan template or earned value management (EVM).

The term “work plan” will be used in this document to refer specifically to a work plan created from the *CxTemp_EarnedValueWorkPlan* spreadsheet. The **Intro** worksheet in the template provides an overview of the work plan spreadsheet. As noted there, although the CxOne work plan template can be used “out of the box”, it is designed to be a powerful spreadsheet as opposed to a software application. This provides the best support for the unique details of each project, but requires some familiarity with Microsoft Excel.

CxSample_EarnedValueWorkPlan is useful to refer to when using this guide.

1.1 Work Plan Purpose

The CxOne earned value work plan and surrounding practices serve several purposes:

- Reduces the amount of effort necessary to effectively manage software projects
- When coupled with effort tracking provides efficient earned value management
- Supports immediate estimate refinement using recent performance data
- Captures bottom-up effort estimates for all project work.
- Models the cone of uncertainty through estimate ranges
- Defines the Work Breakdown Structure (WBS) for all work on the project
- Breaks work into easy to manage weekly iterations, staying above the noise inherent in daily software project activities
- Facilitates use of fuzzy logic for effort and time estimates.
- Provides both summary and detailed views of project performance.

The CxOne earned value work plan is **not**:

> A **detailed schedule** or **scheduling tool**. The work plan is an aggregate, coarse view of all work to complete a project. A detailed schedule is a fine grain view of task scheduling, dependencies, critical path or critical chain management, and resource management necessary to complete some part of a project. Scheduling tools focus detailed schedule creation. The work plan provides critical inputs to detailed scheduling including task breakdown, effort estimates, and time estimates.

Smaller projects using a work plan may be able to manage detailed scheduling informally (e.g, using c*ollaborative construction* and *corrective activity management* techniques).

Larger, more complex projects can use the work plan as an input to incremental detailed scheduling (e.g., using a scheduling tool to create detailed schedules for each delivery milestone based on the task breakdown and estimates in the work plan).

1.1.1 Why Use a Work Plan

The CxOne work plan is designed to address the critical planning and tracking challenges common to software projects. Scheduling tools are commonly employed to plan and track work on software projects. However, their use alone does not guarantee increased schedule or budget predictability nor does it guarantee visibility into the actual state of the project.

A schedule is a model. It is no better than the inputs it is built from. Too often software projects expend effort managing the critical path of a schedule whose model is too detached from reality to be relevant. Eventually it becomes apparent that even if the project is “tracking to schedule”, it may not ship on time or be on budget.

The CxOne work plan brings together a number of techniques and best practices to strike a optimum balance between:

- long-term and short-term focus
- qualitative and quantitative techniques
- precision and accuracy
- formal and informal processes

1.2 Overview

The CxOne work plan does not dictate a methodology or lifecycle. It can be adapted to a wide range of project types and sizes, and has proven extremely valuable for small to medium projects (2-15 people, 1-18 calendar months) using some form of iterative lifecycle.

Best practices built into the work plan include earned value management, bottom-up estimation, time tracking, binary milestones, the cone of uncertainty, and iterative development. The spreadsheet tool and related processes are designed to provide pragmatic support for creation of a high-quality model of a project’s work.

The work plan has significant planning and tracking value by itself and can negate the need for more detailed scheduling on many projects. It can also be used to vastly improve the data used as the input for building detailed schedules with schedule tools. Scheduling tools do not handle the effort uncertainty inherent in software projects very well and tend to encourage a precise, calendar focused view. Using the work plan to model an aggregate view of a project that includes uncertainty while using a scheduling tool to model the expected work plan in more detail provides two valuable perspectives for managing the project.

1.2.1 Guide Contents

The sections of this guide start with material that is generally applicable to project management, and move towards material that is specific to *CxTemp_EarnedValueWorkPlan*. The document sections are:

- Planning and Tracking
- Creating the Work Plan
- Effort tracking

- Managing the schedule
- Spreadsheet Entry
- Using the summary page

1.3 Lifecycle

A project's work plan should be developed as early as possible. This allows for modeling of the WBS as well as early bottom-up estimation. Early in a project, it will only make sense to define the work plan's top levels. As project work progresses the level of detail can be increased for the areas of the work plan the project has visibility into (see ***CxBest_ProjectHeadlights***).

The work plan can be volatile early in a project, with even the top level structure open to change. Although developing the plan early may require some rework, the early drafts of the plan will help to drive planning and estimation work and identify risks early.

The work plan is relevant for the entire life of the project. As the project progresses, the work plan should be updated regularly to keep the work breakdown and estimates accurate. Baselining the work plan to track changes on planning over time is important, but should be balanced with ensuring that the most recent work plan represents the best understanding of the remaining work necessary to successfully complete the project.

1.4 Process

1.4.1 Entry Criteria

Depending on the type of the project and the organization the work initiation process should have the following inputs (or their equivalents):

- Initial draft of project plan
- Initial draft of the requirements and architecture
- Initial draft of the business schedule

1.4.2 Skill Requirements

- Competence to leadership engineering management skills are required to produce the work plan, depending on its scope and sophistication.
- Some competence with Microsoft Excel is required.

1.4.3 Tools / Techniques

- *CxTemp_EarnedValueWorkPlan*
- *CxCheck_TaskEstimation*
- *CxPattern_WorkPackages*
- Sample work plans

1.4.4 Exit Criteria

- The work plan undergoes at least weekly review and update throughout the entire project. It is complete when the project is complete.
- Depending on baselining and project approval processes, revisions of work plans may need to be signed off on a regular basis (such as for each milestone).

1.4.5 Downstream Customers

- Project managers, technical leads, the project team, stakeholders, and customers will use the work plan as a tool for defining project planning for work breakdown, effort estimates, and timing.
- The project team manager uses the work plan as the framework for capturing bottom-up project estimates
- The work plan provides for earned value tracking against the defined plan for all stakeholders

1.4.6 Combination Options

The work plan is usually treated as a singular document for a project. A complex project may be split into separate efforts with different work plans. If detailed schedules are used, they will be tightly coupled to the work plan.

2 Planning and Tracking

This section provides a contextual model for using the CxOne work plan and describes surrounding assumptions, techniques, and ideas that will increase the efficiency and effectiveness of project planning and tracking.

2.1 Planning Levels

The work plan assumes planning and tracking will occur at three levels:

- **Top-Level Project Planning**. This is the top-level aggregate planning for the project. Normally this will be defined through a combination of
    - a project plan that defines project goals, risks, constraints, and processes
    - a description of functionality the software will provide (requirements)
    - a technical plan for executing the functionality (architecture)
    - a business schedule that defines detailed project goals, risks, and constraints
    - a project estimate that defines size, budget, resource, and time estimates

  How detailed and extensive the top-level planning is vs. the downstream planning will depend on where your project lifecycle falls on the spectrum of waterfall vs. evolutionary prototyping. The work plan will reflect the lifecycle choice by providing appropriate levels of detail and uncertainty.
- **Milestone Planning**. Most projects will split the project delivery into iterative milestones. The work plan uses milestones to model overhead work that is not readily assignable to functionally decomposed work packages. Project management and system testing are good examples of activities that are modeled on a per-milestone basis as opposed to being tracked separately for different functional areas of the project. Milestones will often match a project's headlights (see section 2.2).
- **Weekly Iteration Planning**. The work plan breaks the project down into weekly iterations, and update of the work plan will occur on a weekly basis. The planning for each week should be driven by the information in the work plan, and significant changes that are encountered in work breakdown, estimates, or scheduling should be fed back into the work plan. Iterations do not have to be a week in length, but this simplifying assumption works well for most projects.

2.2 Project Headlights

"Planning beyond your project headlights" is the idea that it is only worth planning fine grain task execution within the scope of time that you have good visibility of, i.e., you have a reasonable probability of accurately modeling the work. A one year project starting in January shouldn't be laying out tasks with daily precision for November. However, using daily precision for the upcoming 3 weeks, weekly precision for the next 12 weeks, and monthly precision thereafter is very appropriate. See ***CxBest_ProjectHeadlights*** for more detail.

2.3 Work Breakdown

The work plan captures the Work Breakdown Structure (WBS) for a project. The WBS should be decomposed and modeled as a tree hierarchy. Each leaf of the tree is a work package. Once a work package is scheduled to start, it should be an atomic unit of work. Depending on how the project is managing its headlights though, a work package may start as an estimated aggregation of work that will be broken down once the project has progressed further and better illuminated the work (see section 2.3.1).

The goal of a WBS is to model all work on a project in a way that supports a natural view for managing the work, i.e., the WBS should be easy to map onto the deliverables of the project team. As much as possible, the decomposition of work should also allow easy mapping to system functionality and architecture, but exceptions should be made if a different WBS view will make managing the project easier. It will also often make sense to couple the work plan to methodologies and techniques being used on the project. For example, if functionality is being defined with use cases, breaking the work plan down along use case lines will likely create an easily managed WBS. Common patterns that can be useful for modeling work packages are provided in ***CxPattern_WorkPackages***.

Effort ranges and open/close weeks will be estimated for each work package. The work packages should be optimized for earned value tracking with binary completion, which means they should have clear open and close criteria and occur in contiguous blocks of time.

2.3.1 Detail Level and Size of Work Packages

The work plan defines the detail level that the planning and tracking lead will use for planning, estimating, scheduling, tracking, and controlling the project. Items may be broken down into grater detail for detailed scheduling, individual planning, estimation, or other purposes, but the work plan defines the level at which most project tracking and control occurs.

When using iterative life-cycles, work packages that are farther in the future will normally be larger and less defined than near-term work packages. As the project moves forward work packages can be decomposed into a hierarchy of smaller work packages, thus increasing the level of detail. It is normally best to start a project’s work plan as early as possible with a top-level structure, and then fill out and modify that structure as project planning, requirements, and design work provide greater visibility into the optimal work breakdown.

The optimal size of work packages will be driven by a large number of factors including the nature of the work, the size of the project, and the level of detail the work plan is being used to model. An average work package size for packages that represent high-level roll-ups of future work might range from 50-1000 hours depending on the size, nature, and position in lifecycle of the project. An average work package size for packages that have been fully decomposed may range from 5-100 hours.

It is not unusual to have a wide range of work package sizes, as some items may be small but are critical to call out, and others may be left larger because they don’t have a decomposition that will be useful for managing the work. When creating work packages, the size of the package should be a secondary consideration to whether the work package will be contiguous and have clear start and completion criteria.

Whether the work plan is being used to directly manage project tasks, or being used as an input to a detailed schedule directly affects work package size. When using a detailed schedule in conjunction with a work plan, work packages can be larger, since the detailed schedule breaks down and tracks fine grain details of the work package.

2.3.2 Resource Assignment

Work packages are often aggregations of work by multiple staff and can combine several different types of work. For example a work package to deliver a small use-case may involve a collaboration of team members doing design, construction, and testing. The specific resources don't have to be called out as part of the work plan; the work plan's focus is on aggregate effort, not staffing. The range of estimates assigned to the work packages can be used to capture the uncertainty of staffing future items (e.g., this will take 3-4 hours if Sally does it, but 4-8 hours if Jake does it, so we use 3-8 hours as initial effort estimate).

If you have specific skill sets and know who needs to do what, capture it in the plan. If the team members are cross-functional or have not yet been assigned, assigning specific resources for future work plan items is not critical (the details of exactly who does what when can be worked out when it comes time to perform the work).

2.4 Estimation

Work package estimation is normally a micro estimation exercise that focuses on capturing the **EFFORT** range considered probable for completing all the work defined as being part of that package. The **TIME** estimate to complete the work is separated from the effort estimate, and should be derived from the effort estimate and scheduling issues.

Thus if you had a task you felt could be completed with 10 hours worth of work by one person, the expected effort would be 10 hours. You may complete that task over a series of days however, as you work on other things. There is obviously a relationship between effort and time, in that it usually takes more effort to do a task if you have to start and stop a lot. The amount of ramp up and down time caused by interruption on a task should be captured as part of the estimation range, and will normally come out in the wash across a set of estimates.

The best technique for estimating effort of work packages is to base it on historical size and effort data from analogous projects. Another good technique for estimating work packages is to have the entire development team develop and agree to the estimate ranges together. See the CxOne estimation materials for more ideas and support.

As noted above, the work plan does not normally focus on specific resource assignments for a work package; the goal is instead to accurately model a range of effort estimates for the work.

2.4.1 Estimate Ranges

All estimates have an amount of uncertainty associated with them. In the planning and tracking spreadsheet the uncertainty of effort estimates is captured by using high, low, and expected values to create a range of possible effort.

For calendar time, uncertainty is captured by using a one week precision to model dates.

2.4.2 Fuzzy Logic Estimation

It can be faster and easier to utilize fuzzy logic techniques for initial estimates. This means assigning a relative value like ‘small’ or ‘large’ to tasks. This value is then converted into low, high, and expected effort values for modeling purposes. As a task is better understood, fuzzy logic estimates should be replaced with more detailed estimates. See the work plan spreadsheet’s **FuzzyValue** tab for more information.

2.4.3 How to estimate “Overhead”

To facilitate easier estimating and use of historical data to support future estimates, try to model as much “overhead” work as possible as part of work packages which are creating tangible deliverables (as opposed to modeling the overhead as a separate activity). As an example, when estimating effort to complete work package X, which represents the construction of feature Y, include the amount of effort you expect to spend dealing with project activities that relate to work package X, but may not be considered “construction” of feature Y. For example, reviews, issue management, reading e-mails, lab and build work, design discussions, refactoring, immediate testing and defect fixing can all be rolled into the estimate.

“Overhead” work generally consists of *recurring, interrupt driven,* or *supporting* activities. By capturing as much ongoing effort as possible as part of work packages that create deliverables, overhead categories can be minimized. This simplifies creation of estimates and use of historical data; as long as the way you perform work doesn’t radically change you can focus on what is being created, instead of decomposing the flow of your day into a bunch of unnatural overhead categories.

General overhead categories should be reserved for overhead that cannot be meaningfully allocated to an existing work package, or which apply to rework or other activities on a previously closed work package. In some cases, project leads may want certain activities to be called out (e.g., support for an outside project), in which case work packages should be created to model these activities.

2.4.4 Buffers

Buffers are overhead categories added to the work plan to handle unplanned work or provide for other contingencies. The use of buffers should be minimized and structured in such a way the earned value tracking is not degraded by significant effort falling into buffer categories.

2.5 Effort Tracking

Keeping track of where everyone’s effort goes is crucial to effectively plan and track software projects. Effort should be tracked against work packages, in a symmetrical fashion to how estimates were created. This means that the same assumptions used during estimation of work packages, in terms of what work the package represents, should be applied when determining whether to log effort against a work package.

Precision of effort tracking can be 0.1 hour, 1 hour, or even 1 day granularity depending on project needs. Having individuals keep a personal time log is great, but not necessary. As a minimum everyone should make reasonable estimates of where the effort in their day went (preferably at the end of each day), and enter it into a tracking system.

An effort tracking system can be an elaborate database application, a simple sheet of paper, or anything in between. To use the automated earned value capability of the CxOne work plan, effort entry records need to be imported into the work plan spreadsheet with at least the name, amount of effort, and the WBS tag the effort was related to. Database time tracking can simply import records into the raw time worksheet. Manual systems can cut and paste, or manually type in time records.

If you do not have a readily available time tracking system, you may need to rely on a manual process of coordinating paper or spreadsheets. If done well, such a system does not need to incur a lot of overhead. However, when relying on manual time tracking you may want to limit the granularity of work breakdown to strike a balance between visibility into the work and overhead of processing effort tracking data.

CxTemp_EffortTracking provides a simple spreadsheet template for effort tracking.

2.6 Earned Value Management

Earned value management (EVM) is a powerful technique for measuring project performance and predicting project outcomes. EVM allows for greater visibility and early problem detection. The concepts of earned value management are intuitive and the technique has been around for over 30 years. It has not been widely used on software projects partly due to the complexity of making the simple concepts simple to execute in practice.

Taken as a whole, the CxOne work plan and surrounding ideas described in this guide simplify the complexity associated with using EVM on software projects. In addition to the general project management techniques discussed in this guide, simplifying assumptions for employing EVM with the CxOne work plan are:

- **Effort hours represent cost.** Most software project’s budgets are tied up in the hours staff spend on a project. Even when the cost of this effort is variable (e.g., differences in wages or “free” overtime) or there are significant direct costs (e.g., major software or hardware acquisitions) a project’s success or failure generally hinges on how well the total effort of the project is managed to create a valuable output. The CxOne work plan deviates from traditional EVM by using effort hours instead of dollar cost to model work, making data input and analysis simpler.

- **Work plan instead of detailed schedule.** Too often projects fall into a trap of using EVM on a detailed schedule when the schedule does not reflect reality. Applying EVM at a work package level instead of a detailed schedule task level avoids getting bogged down in unnecessary details.

To use EVM you will need to provide an expected effort estimate (the planned value) and planned completion week for each work package. Effort must be tracked so it can be allocated to the work package (the actual cost). Finally you need to identify the week you start a work package, and the week it completes.

2.7 Binary Work Package Completion

Software projects often have tasks that remain 90% complete indefinitely. Using work packages as binary miniature milestones increases visibility into the actual state of the project. Binary completion means that a work package is not closed until it is agreed that the goals of the work package have been completely met (or some part is being explicitly deferred or otherwise managed elsewhere in the work plan). Work packages should be defined in a way that work can be started (opened) and completed (closed) in a contiguous period of time with clear, agreed upon completion criteria.

How to deal with iterative development and rework are key issues to successfully managing closure of binary work packages. Closing a work package means that the significant core work on an item has been completed to the satisfaction of all technical leads. The item may be revisited before the end of the project (e.g., a defect found late), but the work package is still considered closed.

Planned revisiting and iteration can be modeled by creating phased work packages that occur at different times (e.g., have a Phase 1 and Phase 2 construction of a feature).

Unplanned defect fixing and iteration should be treated as overhead and handled as part of recurring milestone overhead. By definition the amount of such work for any single work package is unknown, so is easier and more accurately modeled as a buffer attached to each milestone. If significant unplanned work is uncovered during the course of the project, the work plan should be updated to better reflect the current understanding of the work.

2.8 One Week Resolution

The planning and tracking of work packages, along with earned value calculation, status reporting, and detailed task coordination is designed to occur on a weekly basis. Weeks are a natural way to break down project work, can be easier to work with than dates, and automatically capture a level of uncertainty in time estimates.

The project manager will work with the project team to plan, track, and coordinate weekly execution of the work packages. It will often make sense to have a standing team meeting to review work plan status at the end or start of each week. The project manager should collect information for planning on an as needed basis during the week, and team members should raise immediate flags when risks to the plan are uncovered.

The specific techniques used for weekly planning and tracking will vary depending on the project. For some projects a detailed project scheduling tool can be used to break down work packages into fine grained tasks and do daily scheduling. Others can use less formal approaches such as the *Collaborative Construction*, *CxTemp_WeeklyIndividualPlan*, or *Extreme Programming*, which rely on collaboration and coordination among team members.

2.9 Milestone Planning and Tracking

Incremental delivery milestones should be defined in the work plan. Depending on project size and needs, milestones would normally be 2-8 weeks long. Larger projects may utilize a hierarchy of milestone releases. Milestones provide for tracking of project progress through delivery of software or other tangibles. They also provide a useful mechanism for modeling ongoing overhead work in discrete work packages.

Planning and tracking occur on a weekly “sliding window” basis, i.e., for each week planning should be updated to look forward as far as detailed work is understood.

Milestones provide a natural opportunity to focus planning activities. Each milestone should be planned as a group effort by the team, and if at all possible planning should start before the previous milestone is complete. Planning will be reviewed weekly by the team and daily by the project manager. The project manager will collect data, analyze the tracking and EV information in the work plan each week, and make any necessary adjustments to planned work.

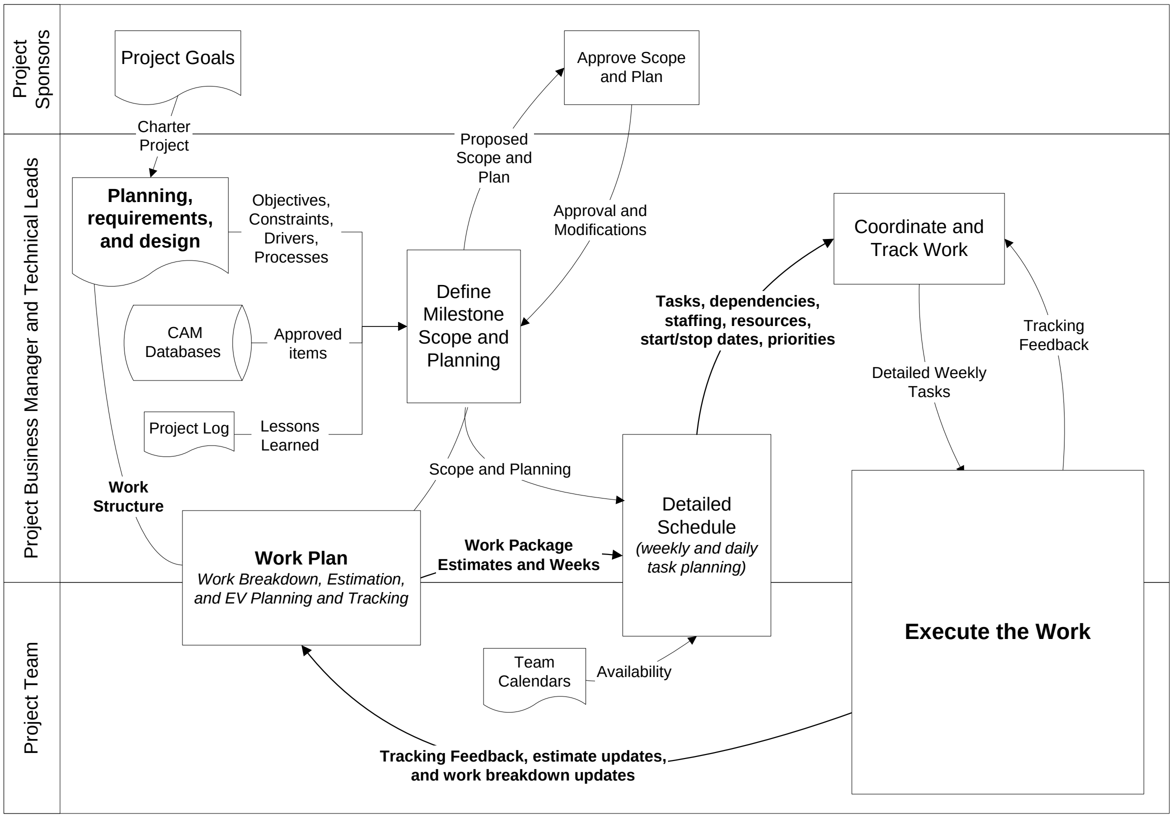

Project Sponsors
Project Business Manager and Technical Leads
Project Team
Project Goals
Approve Scope and Plan
Charter Project
Proposed Scope and Plan
Approval and Modifications
Planning, requirements, and design
Objectives, Constraints, Drivers, Processes
CAM Databases
Approved items
Project Log
Lessons Learned
Define Milestone Scope and Planning
Coordinate and Track Work
Tasks, dependencies, staffing, resources, start/stop dates, priorities
Detailed Weekly Tasks
Tracking Feedback
Scope and Planning
Work Structure
Work Plan
Work Breakdown, Estimation, and EV Planning and Tracking
Work Package Estimates and Weeks
Detailed Schedule
(weekly and daily task planning)
Execute the Work
Team Calendars
Availability
Tracking Feedback, estimate updates, and work breakdown updates

Figure 1 – Example of a milestone planning process

Figure 1 provides a representative workflow diagram for the activities and data that could go into a milestone planning and tracking process.

3 Creating the Work Plan

This section focuses on defining work packages and describes assumptions, techniques, and issues related using the work plan.

3.1 Work Plan Lifecycle

The work plan should be started as early in the project as possible, even if it is only possible to put in place a top-level outline of the work. As the project work proceeds the details of the work plan can be fleshed out, with high-level placeholders being broken down into work packages that will be tracked and controlled. The amount of detail your work plan contains will directly mirror your project lifecycle. A true waterfall would have all work packages completely defined after the design stage is finished, an iterative lifecycle will develop work plan details as the project requirements and design are understood.

In all cases, the work plan should be updated weekly to ensure it accurately reflects the best understanding of the work necessary to complete the project.

3.2 Defining Work Packages

The WBS and ***work packages*** are defined in the **WorkPlan** tab of the spreadsheet. Project work will normally break down into three major types of work:

- **Work that shadows functionality.** This is work specific to delivering particular features that will normally map closely to the architecture, use cases, etc.
- **Fixed project tasks.** Project work and tasks that are unrelated to features or support several areas of functionality (e.g., foundation or infrastructure work).
- **Recurring work.** This sort of work is usually modeled as placeholders for overhead that occurs during each milestone.

All project work will be represented in the work plan. It is important to remember that the work plan is a model, so each work package in the plan may included many specific tasks rolled up into one work package. This section provides useful guidelines for ensuring the context and assumptions surrounding work package definitions are easy to use, consistent, and understood by all participants.

The effectiveness of project planning and tracking will be proportional to the alignment of the project team on what each work package represents.

If team members make different assumptions about what a work package represents, estimates will model different things and completion criteria will be unclear. A major goal of defining the work plan is balancing a pragmatic, easy to use work breakdown that feels natural to the entire team while meeting the visibility needs of the planning and tracking lead.

The work plan defines a model of project work that is useful for various project management techniques. It normally closely shadows technical and project breakdowns (e.g., what needs to be built and what resources are available to build it), but may take a different view of work to better support project management goals.

3.2.1 Guidelines for Work Package Creation

Use the following guidelines and assumptions when creating and estimating work packages. The work package patterns in the next section provide some examples of how these guidelines can be utilized to create reusable work package breakdowns.

- Work packages should attempt to model deliverables such as a project document or software functionality.
- Work packages should be contiguous over a relatively short period of time.
- When modeling work for deliverables take into account as much directly related contiguous work as possible. For example a work package for constructing a software feature should include any requirements, design, quality control, and rework that is expected to occur directly before, during, or after main construction. Some examples include clarifying requirements or UI design, completing technical design, test case creation, reviews, bench and unit testing, correction of issues found by immediate quality control, etc. If there will be significant work performed that is separate from the main activity, then separate work packages should be created.
- In addition to directly related contiguous work, work package estimates should model as much overhead related to the work package as possible. As an example, build integration support should be anticipated as part of the effort estimate for a work package. The general overhead categories used with milestones are reserved for work that either cannot be attributed directly to another work package (e.g., system test or top-level management overhead) and for unexpected work (e.g., we'll set aside X amount of effort each release cycle for general defect work that crop up outside of the planned work).
- Work packages need clearly understood binary closure criteria. For effective earned value management, this completion criteria must be reasonably determinable at the time the work is complete. Otherwise you will have skewed earned value tracking because of delay in verifying completion. As an example, feature based work packages should generally be considered complete when all requirements associated with the feature work package are constructed, bench and unit tested, integrated into the build, and possibly feature tested. Depending on your project this does not necessarily mean that all integration and system testing are complete, that every last issue or defect is taken care of, or that there will be no future work in that area. It simply means that to the best readily available knowledge, the intended work for that work package is complete.
- Rigid rules don't work well for binary closure criteria. A better practice is to require agreement of all technical leads on the closure criteria when the work is being estimated. When work is to be closed, the technical leads should again agree that the closure criteria has been met.
- The size of work packages should be calibrated to fall into a reasonably normal distribution, with the average size of work package being the resolution at which the project manager wishes to track and control the project.

- When estimating work packages, make sure all people participating in the estimate are aware of the assumptions and completion criteria of the work package. This ensures everyone is estimating the same work.
- Work package size and fuzzy logic buckets sizes should correlate well.
- Work packages should be assignable to an individual or coherent group.
- Dependencies between packages should be minimized.
- If a significant amount of work or rework on a work package item is dislocated in time from the original work for the work package, split the work into multiple work packages. For example, build a core feature set in one work package, adding less critical enhancements in a later work package.
- Dependencies in the work plan should reflect MAJOR dependencies. Minor or detailed dependencies should be handled through formal and informal project scheduling and activity coordination.
- Resource assignment in the work plan should reflect MAJOR assignments or responsibilities. More detailed resource assignment is handled through a detailed schedule. A good example is reviews. There will often be a review involved with a work packaged. The total staff time for the review should be included in the estimate, but if someone is participating in the work package only as a reviewer they should not be listed in the work plan as a resource for that task.
- There may be exceptions to rules or assumptions in the work plan if these exceptions make management easier. The project manager will work with the team to ensure that everyone is using the same assumptions to break down work and create estimates.

3.3 Work Package Patterns

There are some major recurring patterns of work package breakdown that can be used to increase the efficiency of project planning and tracking. These patterns are described in ***CxPattern_WorkPackages*** and implemented in ***CxSample_EarnedValueWorkPlan***.

4 Effort Tracking

Effort is the primary expenditure and a scarce resource on most software projects. If you do not keep track of where effort goes, you lack the ability to know where you are at in a project, where you need to go, and how much work is necessary to complete the project.

The CxOne work plan assumes that some form of effort tracking is in place. This section provides a brief overview of work plan issues related to effort tracking and some techniques to consider for effort tracking. For more information on effort tracking see ***CxBest_EffortTracking***.

4.1 What Data does the Work Plan Need?

The work plan spreadsheet is designed to operate with just two pieces of raw effort data. For each record in the **RawEffortData** worksheet you need to collect the:

- Number of staff-hours expended
- WBS ID (from the WorkPlan worksheet)

The spreadsheet is also set up to utilize the name of a person if provided. This is not critical for earned value tracking, but can be very useful for manual reference on the **EffortDetails** worksheet, which is available in the expert tracking view.

The work plan does not care when the effort was expended, that is assumed by the WBS ID. The work plan does not care what the size of the data records are. Thus if you are relying on a manual effort tracking mechanism and want to keep data entry to a minimum, you could aggregate as much effort as you want against a work package in single effort data record.

If you have an automated mechanism for capturing effort you may want to import other effort record values into the spreadsheet for manual analysis. The sample worksheet shows an example of this utilizing a company tracking category in addition to the project WBS category.

4.2 How to Capture Effort Data

The ideal is to deploy an effort or time tracking system in your organization. The system needs to allow users to add a text field (the WBS ID) when they enter their time records. If it supports a standard database query the output can be dumped directly into the **RawEffortData** worksheet. (see MS-Excel help for more details on creating queries).

If you do not have such a system, manual effort tracking can be put in place with little effort and a small amount of overhead. A spreadsheet that captures WBS IDs and hours spent on each work package can be filled out daily or weekly either electronically or on paper. Often most team members will not use more than 5 WBS IDs in a week, so the overhead of record creation is usually minimal. These values can then be cut and pasted or manually entered into the **RawEffortData** worksheet.

5 Managing the Schedule

Some scheduling information is modeled by the CxOne work plan, but providing a detailed schedule is not what the work plan is designed to do. The work plan can be used to directly manage smaller projects with collaborative teams.

For larger projects the work plan serves as an upstream artifact to a detailed schedule. This section discusses using the work plan in combination with a separate detailed schedule.

5.1 Detailed Schedules

When more task visibility and control is needed than the work plan can provide, a detailed schedule can be employed. Detailed scheduling focuses on the near-term part of the project which is "in the headlights". Detailed schedules are often managed in a dedicated scheduling tool like MS Project, Primivera, or Scitor.

A detailed project schedule will likely be based on the work package breakdown, probably further decomposed into tasks. Although a detailed schedule should always be traceable to the work plan, in some cases the detailed schedule view may break tasks down different from the work plan. This requires extra effort, so should only be done if two distinct views of the project work (e.g., a long range view in the work plan, short range view in the detailed schedule) provides a significant benefit.

In addition to work breakdown, expected effort estimates, and weekly timing, the detailed schedule provides detailed dependency and resource information. Timing of work is usually laid out on a daily basis. Further information for performing critical path or critical chain management may also be present, e.g., the earliest and latest dates tasks may be completed.

5.2 Developing Schedule Inputs

When a detailed schedule is used, it often focuses on modeling work for each milestone. All schedule inputs should be developed by the planning and tracking lead with close input from the team.

- Make sure fine-grain dependencies are taken into account. If you plan to work on something Thursday that requires Joe to have something done by Wednesday, make sure this is either represented in a detailed schedule or coordinated among team members.
- As possible, plan your work so that you're working on few concurrent tasks. Try to plan your time so that you can start a task and drive it to completion.
- Ensure that there is a mechanism to determine availability of all team members.
- Do not schedule tasks to span staff absences.

5.3 Tracking the Schedule

The work plan spreadsheet is used to track effort against work package estimates for the aggregate WBS. When a detailed schedule is created, the project manager will track task execution against the schedule on a daily basis, folding data back into the work plan on a weekly basis. A process should be evolved that strikes an optimal balance between scheduling overhead, team effectiveness, project visibility and control, and the project goals.

5.4 Revising the Schedule

If the performance to the detailed schedule falls below an acceptable level the schedule should be revised through the following mechanisms:

- Addressing dependency issues
- Reordering tasks to optimize resource uses
- Reassigning resources for tasks or adding additional resources
- Redefining feature scope, effort estimates, or calendar plan

If it is necessary to significantly redefine feature, effort, or time, normally the work plan will also need to be updated to reflect changes driven by the schedule.

6 Spreadsheet Entry

This section covers the mechanics of entering data into the CxOne earned value work plan spreadsheet. The core spreadsheet functionality makes heavy use of macros, which are more powerful and easier to maintain than formulas for this type of processing. However, there is row, column, and cell location coupling between the worksheets and underlying macros. Before adding or subtracting rows or columns to the spreadsheet you should familiarize yourself with the macros, which have been designed and coded to support easy maintenance.

This section should be read in conjunction with *CxSample_EarnedValueWorkPlan*.

6.1 Work Plan Rows

Each row of the **WorkPlan** worksheet may be either a work package or part of the WBS hierarchy. Work packages are identified by placing an 'x' in the **WP** column. The outline functionality is provided using Excel's grouping functionality, and needs to be set up manually. If you want to use subtotals on row headings, these need to be set up and maintained manually. Once a WBS has stabilized grouping and subtotals are relatively easy to maintain, but early on it is better to leave things less polished until the WBS stabilizes.

Row height and font characteristics are maintained manually. Starting with row 4 the update work plan macro will perform all calculations, fill in fuzzy logic, fill in actual effort data, and do color coding. If the first column is blank, macro processing will skip it.

Normally you'll want to ensure all plan data is up to date by pressing the **Update** button on the **Summary** worksheet. However, if you want to look at local changes to fuzzy logic and color coding you can press the **Update Work Plan** button on the **WorkPlan** worksheet.

6.2 Estimates

6.2.1 Fuzzy Logic

Fuzzy logic estimates allow for easier and systematic estimation of future work. Fuzzy logic can be used for both effort and time in the work plan. The average size of fuzzy logic buckets and the size of work packages need to be loosely correlated for the technique to work well.

To use fuzzy logic, first fill out hourly bucket ranges with labels in the **FuzzyLogic** worksheet. For effort, place labels from the fuzzy logic effort table into the **Fuz Sze** column. For time, place labels in the **Week WP Opened** column. When you update the work plan or spreadsheet, the work plan will be populated with appropriate values from your fuzzy logic table. Blank out the **Fuz Sze** or **Week WP Opened** columns if you want to enter manual effort estimates or week estimates.

6.2.2 Manual Effort Estimation

Work package estimates will often start with fuzzy logic effort values. As work is better understood estimates will migrate from fuzzy logic values and into more precise effort ranges that are entered manually.

When not using fuzzy logic, blank the **Fuz Sze** column and enter low, high, and expected values manually. Note that you can adjust the profile of the distribution in this manner, skewing the expected probability more towards the low or high end of the range if appropriate. Fractional hours are not used for estimates, round up or down as appropriate. Note that this includes using 0 hours for the low end of small estimated efforts.

The **Es Lv** column stands for estimate level. This column is not used programmatically, but can be used manually as a way to indicate the confidence of an estimate. As an example, a simple scale of a, b, and c can be used to indicate high, medium, or low confidence in an estimate. Generally any estimate being done with fuzzy logic is likely of low or medium confidence; if you have higher confidence you should provide the numbers directly.

The people likely to be responsible for a work package should participate in the development of its estimate. When estimating individual work packages, engineers should account for the potential overhead activities that will occur during that task. For example, dealing with issue management databases, reading e-mail on a topic, etc. should be factored into the estimate for each specific work package. Estimates should be revisited regularly and updated if necessary, especially if staffing changes. If requirements or design changes or becomes better understood, changes to the WBS may be necessary to best support estimating the work.

6.2.3 Time Estimation

Each work package should have a planned start (open) and stop (close) week associated with it. This will be used in conjunction with the effort estimates to perform earned value calculations. When not using fuzzy logic for time, enter the week number that you expect to start work on each work package ("open it") and the week you expect to complete it ("close it") in the **Planned WP Open** and **Planned WP Close** columns.

6.3 Opening and Closing Work Packages

6.3.1 Opening Work Packages

To indicate that a work package has started (been opened), enter the week number the package was started into the **Week WP Opened** column.

6.3.2 Closing Work Packages

Work packages are closed when they are agreed to be complete. By closing a work package you earn the value of the package for EVM (the value is the expected effort estimate). To close a work package enter the week in the **Week WP Closed** column.

6.3.3 Weekly Work Packages

Normally, you get the full earned value for a work package when you complete it. For recurring overhead work packages (e.g., the packages of the recurring milestone pattern) that you manage daily this can be difficult to manage in the work plan. However, if you place a ‘w’ (for “weekly”) in the **Week WP Opened** column, it designates that the work package’s earned value should be calculated as spread over the planned open and close weeks. Since earned value for weekly work packages is provided based on time, this feature should be used with discretion and only on true overhead activities that are composed of many ongoing small activities or tasks. When completed, place the close week in the **Week WP Closed** column.

6.3.4 Lack of Visibility Due to Lagging Closure

Two of the CxOne work plan simplifying assumptions that make EVM easier, weekly resolution and binary work package completion, can decrease visibility into the current state of the project when there are many pending hours built up for open work packages that are not yet closed. Normally this distortion is not significant, but it can be noticeable if work packages are not small (relative to the size of the project) and closed frequently. If there are significant work packages occurring over an extended period of time (i.e., >2 weeks for a 12 week project) it can be broken into separate work packages to provide better visibility and to allow earned value can be assigned to the work that has been completed.

Binary work package completion provides benefits that normally outweigh any negatives of tracking distortion. However in some cases if you are confident of the status of a large work package, it may make sense to provide a percentage complete to support incremental earned value tracking. To indicate percent complete for a work package, enter a value between 0.0 and 0.99 into the **Week WP Closed** column. Once the work package is 100% complete, replace the percentage with the week the package was closed.

6.4 Other Functionality

6.4.1 Setting a Work Plan Baseline

The work plan includes one baseline for saving the expected effort, and planned work package close week. You can update this baseline by pressing the **Set Baseline** button. This baseline capability only works well with a relatively stable WBS and estimates. It is assumed that for more robust baseline tracking that versions of the spreadsheet will be branched for each milestone release to allow comparison of current plan to previous plans.

6.4.2 Non-Programmatic Fields

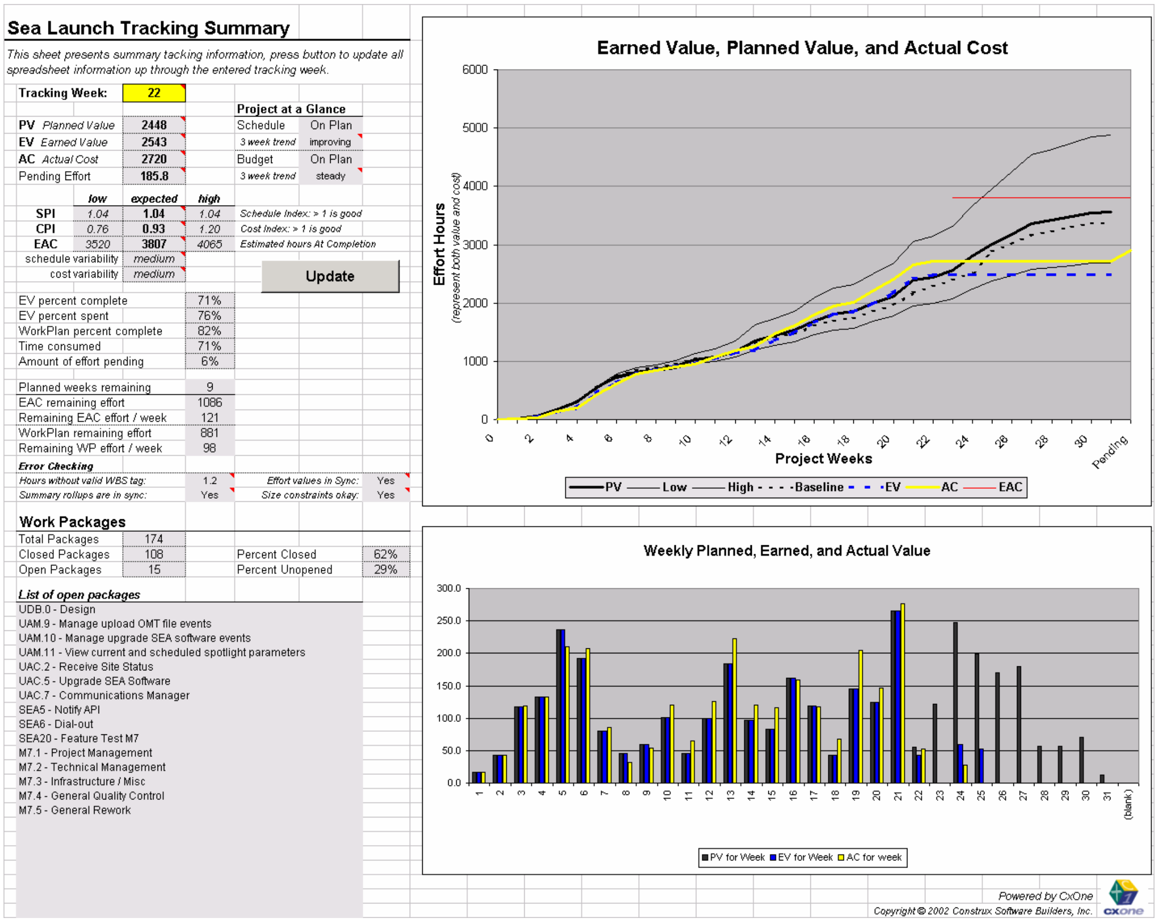
Some fields in the **WorkPlan** worksheet are not used elsewhere in the spreadsheet and are manually maintained by the user. These include **Es Lv** and **Description**. You can enter values in these fields to keep useful information in the work plan or for export to other tools. You can add columns after the description column without updating macros.

6.4.3 Additional Effort Values

There may be cases where you want to apply effort values directly against work packages, without utilizing the raw effort data. In the expert tracking view you can enter effort directly against work packages. This can be especially useful to backfill actual values for early work packages if the work plan was not put into place at the beginning of the project. Negative values can also be used here if it is necessary to permanently adjust time values being queried from a time tracking system that has incompatible records.

7 Using the Summary Page

The CxOne work plan has a tracking summary page which allows critical planning and tracking information to be conveyed in a glance. A snapshot of the summary screen from *CxSample_EarnedValueWorkPlan* is shown below.



Sea Launch Tracking Summary

This sheet presents summary tacking information, press button to update all spreadsheet information up through the entered tracking week.

Tracking Week:	22			
			Project at a Glance	
PV *Planned Value*	2448		Schedule	On Plan
EV *Earned Value*	2543		*3 week trend*	improving
AC *Actual Cost*	2720		Budget	On Plan
Pending Effort	185.8		*3 week trend*	steady

	low	*expected*	*high*	
SPI	*1.04*	**1.04**	*1.04*	*Schedule Index: > 1 is good*
CPI	*0.76*	**0.93**	*1.20*	*Cost Index: > 1 is good*
EAC	*3520*	**3807**	*4065*	*Estimated hours At Completion*
schedule variability		*medium*		
cost variability		*medium*		

Update

EV percent complete	71%
EV percent spent	76%
WorkPlan percent complete	82%
Time consumed	71%
Amount of effort pending	6%
Planned weeks remaining	9
EAC remaining effort	1086
Remaining EAC effort / week	121
WorkPlan remaining effort	881
Remaining WP effort / week	98

Error Checking

Hours without valid WBS tag:	1.2	*Effort values in Sync:*	Yes
Summary rollups are in sync:	Yes	*Size constraints okay:*	Yes

Work Packages

Total Packages	174		
Closed Packages	108	Percent Closed	62%
Open Packages	15	Percent Unopened	29%

List of open packages

UDB.0 - Design
UAM.9 - Manage upload OMT file events
UAM.10 - Manage upgrade SEA software events
UAM.11 - View current and scheduled spotlight parameters
UAC.2 - Receive Site Status
UAC.5 - Upgrade SEA Software
UAC.7 - Communications Manager
SEA5 - Notify API
SEA6 - Dial-out
SEA20 - Feature Test M7
M7.1 - Project Management
M7.2 - Technical Management
M7.3 - Infrastructure / Misc
M7.4 - General Quality Control
M7.5 - General Rework

Earned Value, Planned Value, and Actual Cost
Effort Hours
(represent both value and cost)
0
1000
2000
3000
4000
5000
6000
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 Pending
Project Weeks
PV
Low
High
Baseline
EV
AC
EAC
Weekly Planned, Earned, and Actual Value
0.0
50.0
100.0
150.0
200.0
250.0
300.0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 (blank)
PV for Week
EV for Week
AC for week

Powered by CxOne
Copyright © 2002 Construx Software Builders, Inc.

7.1 Tracking Week

The tracking week is updated manually and contains the week which you want earned value calculations to be done against. This is normally the current week of the project, but can be set in the past to analyze trends.

The information on this page is updated automatically by macros when the **Update** button is pressed. The update process updates all worksheets in the spreadsheet (with the exception of the **WpReport**). The update processing can take a significant amount of time for large projects, so only press when needed. Normally you will update once a week to reflect updates to the work plan, opening and closing of tasks, and newly entered effort data.

7.2 Quadrant Descriptions

The page is split into quadrants. The upper left provides numerical earned value and other tracking information. The upper right displays a classic earned value management chart of the project's progress. The lower left contains metrics about the project's work packages along with a summary of currently opened packages. The lower right provides a different view of planned and earned value that allows for aggregate resource leveling.

7.2.1 Upper Left – Tracking Summary

This area provides critical earned value metrics for the project along with some user-friendly translations. There are comment notes on several fields that describe the meaning of the values. Some blank space is left here to allow projects to add their own summary metrics.

The **Project At a Glance** area provides a user friendly view of the cost and schedule performance, along with an analysis of whether the performance of the previous three weeks was above or below the current cumulative performance.

The **Actual Cost** field contains all hours logged against closed work packages up through the tracking week. The **Planned Value** field contains the expected effort estimate for all work packages that are planned to be closed on or by the start of the tracking week (and any packages closed during the tracking week. The **Earned Value** field contains the expected effort estimate for all work packages that were closed up through the tracking week.

Pending Effort represents effort hours that have been entered against open work packages that are not yet closed. If the pending effort is large relative to the project size the EV analysis may not provide good visibility into the recent performance of the project. You may need to break down the work packages differently (to allow them to close sooner) or into smaller pieces to get better visibility into the state of the project. You can also use percentage completion for some work packages if you are confident in their completion state.

Two key metrics are the **CPI** (Cost Performance Index) and **SPI** (Schedule Performance Index). The CPI is the ratio of earned value to actual cost. If the CPI is greater than 1.0, it means the project is ahead of budget, if it is less than 1.0 the project is behind budget. If the SPI is greater than 1.0 it means the project is ahead of schedule, if it is less than 1.0 the project is behind schedule.

These values are for the aggregate rollup of the entire project; individual work packages may be ahead or behind schedule or budget. To view the performance of individual work packages, use the **EvDetails** worksheet that is visible when selecting the **Expert Tracking View** button on the **Intro** worksheet.

The **Cost** and **Schedule Variability** fields provide a qualitative interpretation of the standard deviation of the cost performance and work package open and close performance for all the individual work packages. If there is significant deviation from the mean among cost or schedule performance values, this field will read *medium* or *high*. This is an indication that the performance to plan of individual work packages is varying widely.

The **EAC** (Estimate At Completion) is a budget estimate for the total number of hours the project will require, calculated by extrapolating the current CPI trend for the remainder of work on the project.

The SPI, CPI, and EAC values are also shown when the actual effort is calculated against the low and high effort estimates.

There are some further metrics that compare percentage complete, percentage used, and remaining for both values based on EV analysis and values from the **WorkPlan** worksheet.

In the error checking section the number of hours that do not have correct WBS tags are noted here – these hours will not be part of the earned value calculations. There are also some error checking formulas on this page that help ensure the manually maintained work plan information is in sync, the spreadsheet is working correctly, and the amount of data in the spreadsheet is within the size constraints of the spreadsheet.

7.2.2 Upper Right – EVM Chart

This graph shows the performance over time of the project. The cumulative value for PV, AC, and EV are charted for each week. The low and high PV are shown, providing the cone of uncertainty for the project's effort estimates. If a baseline was set in the work plan, the PV for the baseline is shown.

The current week is indicated by where the AC and EV lines start to run flat to the right edge of the graph. This allows a clear view of the difference between the current AC and EV. The far right column of this graph will always capture 'Pending' effort. The AC line will normally tick up in this column, representing work that is underway but for which value has not yet been earned.

7.2.3 Lower Left – Work Package Summary

This section provides summary metrics on the state of work packages along with a list of currently open packages.

7.2.4 Lower Right – PV Leveling Chart

This graph provides similar data to the EVM summary chart, but in a view that shows weekly values instead of cumulative values and is based on the planned close week instead of the actual close week.

The primary purpose of this chart is to support filling out weekly estimates for leveling the planned value for the entire project. It also provides a view of the performance to plan on a per week basis.

7.3 General Analysis Issues

7.3.1 Volatility from Small Data Samples

Since calendar resolution is kept to one week, and the work plan focuses on aggregate planning relationships, the work plan summary information can be volatile with small work plans or at the early stages of a project.