

Chapter XIV

Performance Reporting and Earned Value Analysis

Never mistake motion for action.

(Ernest Hemingway)

A PM must report on the project's performance to the upper management of the performing organization and perhaps also to the benefiting organization. Upper management usually realizes the complexity and rapid dynamics of IT projects, however, it still needs accurate projections of project completion time and cost so that interdependent business activities, including product release/migration can be planned. Traditional methods of progress-performance reporting are often inaccurate and misleading. Earned value analysis (EVA) has proven to be an extremely effective tool for project time and cost management providing good estimates of actual project completion cost and date. EVA is also a good early indicator of project problem areas, so that appropriate corrective action can be initiated. The effective use of earned value in IT projects, however, is still low, particularly outside of the U.S. government and its contractors.

The application of EVA in IT projects is neither trivial nor straightforward because earned value requires careful planning and reporting in regard to work-packet specification, progress measurements, and actual cost determination. In addition, mechanisms to obtain all this information must be integrated into the organization's project management, software engineering methodology, and financial reporting systems. This chapter discusses and illustrates effective ways to integrate EVA into an organization's methodology and financial systems, and also presents specific techniques to deal with associated EVA complexities.

Traditional Performance Reporting

A PM needs to track and review progress and cost, but should not intervene directly unless there are problems. When there *are* problems, early intervention is more effective than delaying corrective actions; thus the PM needs methods and systems that not only accurately and efficiently track cost and progress but that also highlight problems and pinpoint the trouble spots. The problem is that traditional project progress reporting is neither accurate nor effective in the early identification of problems, nor in projecting cost or time to complete.

For illustrating traditional reporting problems and the merits of EVA, I will use a simple yet effective example from previously published works (Brandon 1998, 1999). An earlier chapter discussed the formation of the WBS for a project; and that formulation involves subdividing the work to be done into tasks, or work packets, which are organized into a logical pattern.

For this example, the WBS consists of Level 0 and Level 1, shown graphically in Figure 14.1 for Level 0, and for both levels in the list format of Figure 14.2. As discussed in an earlier chapter, the top level of the WBS is sometimes called Level 1, but in IT, the top level is usually numbered Level 0.

As discussed in an earlier chapter, in IT the work definition is usually based on the organization's software development methodology. Any type of methodology can be used (waterfall, overlapping waterfall, spiral, iterative, incremental, etc.), but this example uses a classical waterfall methodology. Iterative and incremental methodologies can be more difficult to schedule and employ EVA because one does not know in advance how many increments or iterations may be necessary. However, a schedule and EVA cost plan can be set up for each iteration (at the beginning of that iteration). Alternatively, a "bounding box" methodology can be used. As was discussed earlier in the book, a time or cost boundary is established for the phase of the project being scheduled and costed.

Typically, each work packet is assigned to an organization at the upper WBS levels and to individuals at the bottom WBS levels. The organizational structure may also be

Figure 14.1. Example WBS (Level 0)

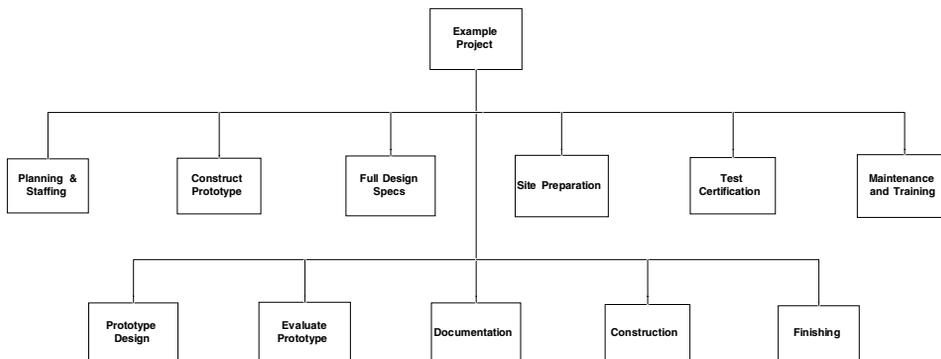


Figure 14.2. Example WBS

| <u>Level Zero</u> | <u>Level One</u> |
|--------------------------|---|
| Planning & Staffing | Project Plan Resource Commitments |
| Prototype Design | Requirements Design |
| Construct Prototype | Detail Design Construction |
| Test/Evaluate Prototype | Test Plan Development Component Testing Full Testing Destructive Testing Test Documentation |
| Full Design Specs | External Specifications Internal Specifications |
| Documentation | Internal Specifications Required Documents |
| Site Preparation | Site Design Site Construction |
| Construction | Component Construction Component Assembly |
| Test/Certification | Test Plan Development Component Testing Full Testing Destructive Testing Test Documentation |
| Finishing | Component Finishing Assembly Finishing |
| Maintenance and Training | Maintenance Personnel Supervisory Personnel |

represented in a hierarchically manner typically called an organizational breakdown structure (OBS). The amount and type of cost to complete each work packet is then estimated, and then those costs are rolled-up to the higher WBS levels. For IT projects, most packets involve only labor and the amount of work can be estimated by any of a number of techniques described earlier in the book. Resources to perform the work are identified either generically or specifically for each packet and may be coded by using a resource breakdown structure (RBS). Each packet also typically has one type of cost (labor, travel, materials, etc.), coded by an element of cost (EOC) breakdown or general ledger (GL) account number. If the full GL account number is used, reconciliation to the organization’s accounting system may be easier.

Thus, each work packet is the intersection of these coding dimensions, and it has a detailed objective and work description plus a specification of WBS, OBS, RBS, EOC, estimated cost, and dependent tasks. “The work package usually has a short-time span schedule (40 to 100 hours), is limited to one performing department, and has defined completion criteria” (Kiewel, 1998). When these tasks are “rolled-up” the WBS hierarchically, the total cost plan is derived as shown in Figure 14.3 in spreadsheet form for the example project herein. This is also illustrated graphically in Figure 14.4.

In developing the cost plan using a bottom-up estimation method, each Level 0 WBS item would be further broken down into another spreadsheet that shows the estimated cost

Figure 14.3. Project cost plan in spreadsheet

| Project Cost Plan | | | | | | | | | | | | | |
|----------------------|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-------|
| | Ja | Fe | Ma | Ap | Ma | Jn | Jl | Au | Se | Oc | No | De | Total |
| Planning & Staffing | 3 | 2 | | | | | | | | | | | 5 |
| Prototype Design | | 3 | 3 | | | | | | | | | | 6 |
| Construct Prototype | | | 8 | 8 | | | | | | | | | 16 |
| Evaluate Prototype | | | | 5 | 10 | | | | | | | | 15 |
| Full Design Specs | | | | | 5 | 6 | 3 | | | | | | 14 |
| Documentation | | | | | | 2 | 2 | 1 | 1 | 1 | 1 | | 8 |
| Site Preparation | | | | | | 8 | 3 | 3 | | | | | 14 |
| Construction | | | | | | | | 20 | 50 | 50 | 20 | | 140 |
| Test/Certification | | | | | | | | | 10 | 6 | 4 | | 20 |
| Finishing | | | | | | | | | | | 8 | 4 | 12 |
| Maintenance Training | | | | | | | | | | | | 4 | 4 |
| Monthly Cost | 3 | 5 | 11 | 13 | 15 | 16 | 28 | 54 | 61 | 27 | 13 | 8 | 254 |
| Cumulative | 3 | 8 | 19 | 32 | 47 | 63 | 91 | 145 | 206 | 233 | 246 | 254 | |

Figure 14.4. Project cost plan graph



Figure 14.5. Cost plan for prototype

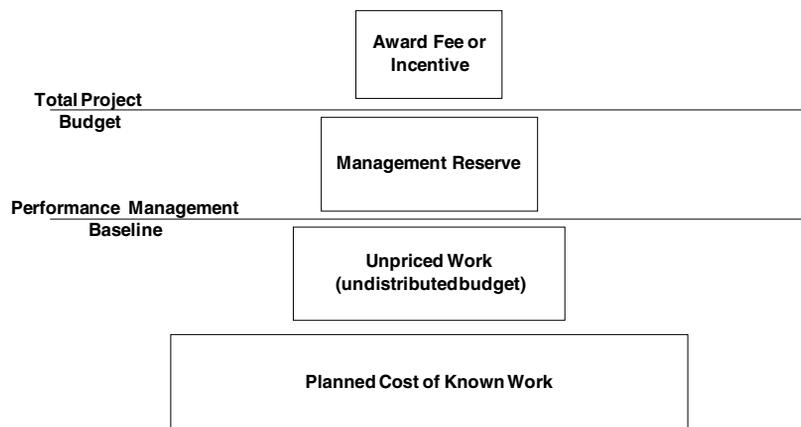
| Test/Evaluate Prototype | | | | | | | | | | |
|-------------------------|----------|-----------|----------|----------|----------|----------|----------|----------|----------|-----------|
| | A | M | J | J | A | S | O | N | D | Plan |
| Test Screens | 3 | | | | | | | | | 3 |
| Test Reports | 2 | | | | | | | | | 2 |
| Test Processing | | 2 | | | | | | | | 2 |
| Audit Trail Verify | | 4 | | | | | | | | 4 |
| Destructive Testing | | 4 | | | | | | | | 4 |
| Monthly Plan | 5 | 10 | 0 | 15 |

and schedule for associated the level one work packets, as shown in Figure 14.5 (for one of the level zero items).

This cost plan is sometimes called the *planned cost based on known work* and may include risk factors at the individual WBS work-packet level. The inclusion of added cost based upon risk factors at the WBS work-packet level (mainly estimation risk factors) and overall project risk was discussed in Chapter VIII. There may also be some unpriced additional (real or suspected) work added which brings the cost up to the expected work cost or the performance measurement baseline (PMB). The difference between the expected work cost and the planned cost based upon known work is called the “undistributed budget.” The undistributed budget is for *known work* and sometimes used as a holding account until a formal contract change is made (and new WBS codes are added).

The project manager generally has responsibility to manage to the performance measurement baseline. In addition a management reserve may be added as a reserve for overall project risks, including unforeseen work that is still within the overall scope of the project. The performance measurement baseline plus the management reserve adds up

Figure 14.6. Cost breakdown in contracts



to the total project budget (or, for a contract situation, the negotiated cost). This is illustrated in Figure 14.6.

In contracting situations, the total contract price for a project will include a profit for the contractor. The performance measurement baseline normally excludes the profit for performance calculations (i.e., EVA) on both the part of the contractor and the buying organization in the case of cost-plus-fee contracts. In time and material contracts it is normally included, because it is typically built into the rate structure of the resources.

As the project progresses, actual costs are incurred by the effort expended in each work packet. Hopefully, these actual costs are measurable at least in total, as they may not be measurable for each work packet at the lower levels of the WBS (this is discussed later). However, the relative amount of things to be accomplished within the work packet that has actually been completed (percentage complete) can be estimated. The planned cost may change in time also (change orders), and thus work packets may need to be reestimated or new ones added or deleted. Figure 14.7 shows an example of a Microsoft Project entry screen for entering progress information. In Microsoft Project, cost information cannot be directly entered (without altering system defaults), but it is typically input via resource utilization, where resources have been given a particular rate. For tasks that span multiple progress reporting time periods, Microsoft Project generally assumes a linear distribution of both planned and actual costs across the task (although some front-loading or back-loading contouring patterns may be used).

Figure 14.8 shows an example of the Web-based FiveAndDime screen for entering progress information and cost data (a progress transaction). FiveAndDime has other screens for simplifying the labor entry process, illustrated later in this chapter.

There are several types of traditional performance reports. One type is status reports that describe the time and cost spent on each WBS item rolled up to the reporting level. The reporting level is typically Level 0 or Level 1. These may be displayed in a list format, or in a WBS tree diagram showing the person hours and/or dollars spent for each item. The progress report is a very popular report that describes what has been accomplished in terms of percentage complete by task. This information is typically shown superimposed

Figure 14.7. Task form in Microsoft Project

| Task Information | |
|---|---|
| General | Predecessors Resources Advanced Notes |
| Name: | Evaluate Prototype Duration: 45d <input type="checkbox"/> Estimated |
| Percent complete: | 80% Priority: 500 |
| Dates | |
| Start: | Tue 4/1/97 Finish: Mon 6/2/97 |
| <input type="checkbox"/> Hide task bar | |
| <input type="checkbox"/> Roll up Gantt bar to summary | |
| Help | OK Cancel |

Figure 14.8. Progress transaction form

The screenshot shows a web browser window titled "Add New Entry to Database - Microsoft Internet Explorer". The main heading is "Add New Transaction". The form contains the following fields and controls:

- WBS Code:
- Period Name:
- Cost:
- Resource:
- Hours: (enter either cost, or resource and hours)
- % Complete:
- Notes:
- Submit:

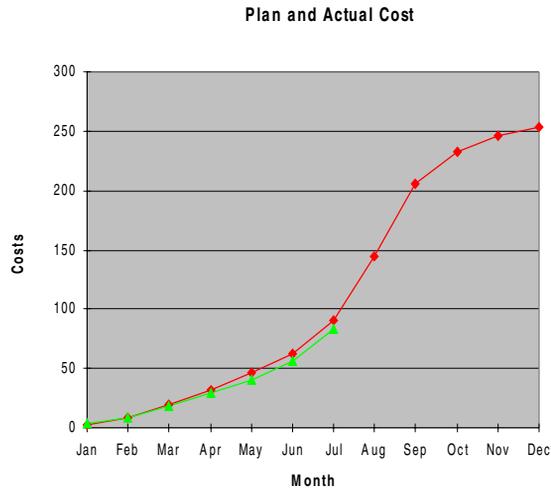
on the Gantt chart as percentage complete numbers inside of the bars or shading a portion of each bar in an amount proportionate to the percentage-complete value. An example of this is shown in Figure 14.9. There are also variance reports that compare actual values to planned values, usually in terms of cost either measured in currency or person-hours. This is usually reported for the entire project or at Level 0. Trend reports show metrics as a function of time, and common metrics to plot are planned costs, actual costs, or overall percentage complete. Figure 14.10 is an example of this for planned and actual cost in currency (dollars). Forecasting reports extrapolate trend curves.

For most organizations using traditional progress reporting, two types of displays are presented to management: progress data and cost data. Progress data are typically shown in a Gantt or similar type chart, as is shown in Figure 14.9 for the example project here.

Figure 14.9. Gantt chart



Figure 14.10. Planned and actual cost plan graph



Cost data are typically reported as actual cost for the entire project versus planned cost, as is shown in Figure 14.10 for the example here. In that figure the planned cost curve is marked with the diamond symbols, and the actual cost curve is marked with the triangle symbols.

The problem with these usual methods is that they do not provide a clear quantitative picture of the true project status, nor do they provide a quantitative means for extrapolating project cost to complete or completion date. *For the example project here, consider the Gantt chart above (which also shows the task percentage complete as dark bars stripes inside the bars) and the actual cost versus budget graph. The project is not over budget (actual cost of \$83,000 versus planned cost of \$91,000), however it is hard to determine if it is on schedule or not. Actually this project is well behind schedule and is over-spent as the forthcoming earned value analysis will show.*

Earned Value Analysis

Earned value (EV) is basically the value (usually expressed in currency) of the work accomplished up to a point in time, *based upon the planned (or budgeted) value* for that work. The original U.S. government's term for earned value is *budgeted cost of work performed* (BCWP). EVA (or EVM—earned value management) is a progress-versus-plan-based metric to evaluate the true performance of a project in terms of both cost deviation and schedule deviation. It also provides a *quantitative* basis for estimating actual completion time and actual cost at completion; EVA can provide an early warning

of project time and/or cost problems as early as 15% into the project (Fleming & Koppelman, 1998).

The EV concept has been around in several forms for many years, dating back to cost variance metrics defined in the 1950s. In the early 1960s, program evaluation review technique (PERT) was extended to include cost variances and the basic concept of earned value was adopted therein. PERT did not survive, but the basic earned value concept did. That concept was a key element in the 1967 U.S. Department of Defense (DoD) policy called “cost/schedule control systems criteria” (C/SCSC). Early implementations of C/SCSC met with numerous problems, the most common of which was overimplementation due to excessive checklists, data acquisition requirements and other paperwork, specialist acronyms, and overly complicated methods and tools. “Much of the criticism that has been levied on earned value in the past was due to an inflexible and dogmatic approach that was applied in the early days” (Webb, 2003). However, after several decades of refinement, C/SCSC is now very effective and the U.S. government has accumulated many years of statistical evidence supporting it. Also in the last decade, initiatives within DoD have been successful in removing excessive and ineffective components of the C/SCSC (Abba, 2000). The C/SCSC have now been successful, and have met the test of time for nearly 3 decades on major government projects (Abba, 2000).

EVA methods have migrated and evolved throughout the world and are now internationally recognized as an effective project management tool. However, these methods have not been widely applied in IT projects (Christensen & Ferens, 1995). Also, until recently EVA use was still minimal outside of the U.S. government and its contractors (Fleming & Koppelman, 1996). There are several reasons for this continued lack of use, including the overcomplication of some surrounding methodologies and procedures, the human factors involved in gathering the necessary input data, the careful planning and WBS setup required, and integrating of EVA effectively with other management information systems. Methods for overcoming these obstacles in general were discussed in the *Project Management Journal* (Brandon, 1998), and such effective methods for IT projects are discussed later in this chapter.

Earned value terminology was originally formalized in DoD standard 7000.2. The ANSI standard (ANSI/EIA-748-1998) on earned value management systems (EVMS) was approved in May of 1998. In 1999 the DoD adopted the ANSIEVMS Standard for use on defense acquisitions. Other governments and organizations have adopted similar standards for earned value such as the Canadian General Standards Board (CGSB 187.1-93 November 1993). These EVMS guidelines incorporate best business practices for EVA systems that have proven to provide solid benefits for enterprise planning and control. In EVA, the three key terms and definitions are:

- *BCWS*: Budgeted cost of work scheduled (the cumulative planned cost) as of a certain point in time
- *ACWP*: The cumulative actual cost of work done as of a certain point in time
- *BCWP*: This is the “earned value” and is determined from the product of the budgeted cost of each work packet times the percentage complete of each work packet as of a certain point in time

Figure 14.11. EV calculations in spreadsheet

| Earned Value—Through July | | | | | | | | | | | | |
|---------------------------|---|---|----|----|----|----|----|-----|------|-----|------|--|
| | J | F | M | A | M | J | J | ... | Plan | % C | EV | |
| Planning & Staffing | 3 | 2 | | | | | | | 5 | 100 | 5 | |
| Prototype Design | | 3 | 3 | | | | | | 6 | 100 | 6 | |
| Construct Prototype | | | 8 | 8 | | | | | 16 | 100 | 16 | |
| Evaluate Prototype | | | | 5 | 10 | | | | 15 | 80 | 12 | |
| Full Design Specs | | | | | 5 | 6 | 3 | | 14 | 50 | 7 | |
| Documentation | | | | | | 2 | 2 | | 8 | 12 | 0.96 | |
| Site Preparation | | | | | | 8 | 3 | | 14 | 36 | 5.04 | |
| Construction | | | | | | | 20 | | 140 | 15 | 21 | |
| Test/Certification | | | | | | | | | 20 | 0 | 0 | |
| Finishing | | | | | | | | | 12 | 0 | 0 | |
| Maintenance Training | | | | | | | | | 4 | 0 | 0 | |
| Monthly Plan | 3 | 5 | 11 | 13 | 15 | 16 | 28 | ... | 254 | | 73 | |
| Cumulative | 3 | 8 | 19 | 32 | 47 | 63 | 91 | ... | | | | |
| Monthly Actual | 4 | 4 | 10 | 11 | 12 | 15 | 27 | 0 | 83 | | | |
| Cum. Actual | 4 | 8 | 18 | 29 | 41 | 56 | 83 | | | | | |

For this example, Figure 14.11 shows the EVA calculation through July. Note that the calculation of EV does not involve actual costs, but is simply the product of the percentage complete and the planned cost.

Percentage complete is determined for each work packet and then EV is calculated for each packet. The total project earned value at a point in time is determined by a WBS rollup of the EV values. For example if a packet had an estimated total cost of \$10,000, and if the things to be done in the packet were 70% complete (or 70% of the things were done), then the earned value would be \$7,000. A WBS rollup example is shown in Figure 14.12, for the level one tasks corresponding to the Level 0 item of evaluate prototype.

Variances between the three values BCWS (planned cost), BCWP (earned value), and actual cost (ACWP) yield the earned value metrics. There are earned value metrics available for both cost and schedule variances. The cost metrics are:

- Cost Variance (Dollars) = BCWP - ACWP
- Cost Variance (%) = (BCWP - ACWP) * 100/BCWP

Figure 14.12. EV calculation for prototype activities

| | Ap | Ma | ... | Plan | % C | Value |
|---------------------|----|----|-----|------|-----|-------|
| Test Plan Devel. | 3 | | | 3 | 100 | 3 |
| Unit Testing | 2 | | | 2 | 100 | 2 |
| Full Testing | | 4 | | 4 | 100 | 4 |
| Destructive Testing | | 4 | | 4 | 50 | 2 |
| Test Documentation | | 2 | | 2 | 50 | 1 |
| Monthly Plan | 5 | 10 | 0 | 15 | 80 | 12 |

- Cost Performance Index (CPI) = $BCWP/ACWP$
- Estimated Cost at Completion (EAC) = $ACWP + (BAC - BCWP)/CPI$
- Or in its simpler form, $EAC = BAC/CPI$

Where BAC is the budget at completion, ACWP is a sunk cost and (BAC - BCWP) is the estimated remaining work. The CPI is also called the cost efficiency factor; values less than 1 indicate that the project is costing more than planned, or that one is getting CPI of a dollar for each \$1 spent. This is good prediction formula if one expects that future performance will be similar to past performance. There are several other (EAC) formulas, and the most appropriate depends upon project type, whether past performance is a good indicator of future performance, and when in the project the EAC is calculated (Christensen et al., 1995). The PMI PMBOK (PMI, 2000) provides this formula when past performance is not expected to be a good indicator of future performance:

$$EAC = BAC - (BCWP - ACWP),$$

which is the budget at completion adjusted for the cost variance. Here one is assuming that corrective action will be effective so that the poor performance of the past will not continue and future performance will follow the original plan.

The schedule metrics are:

- Schedule Variance (Dollars) = $BCWP - BCWS$
- Schedule Variance (Months) = $(BCWP - BCWS)/(Planned\ Cost\ for\ Month)$
- Schedule Variance (%) = $(BCWP - BCWS) * 100/BCWS$
- Schedule Performance Index (SPI) = $BCWP/BCWS$
- Estimated Time to Complete (ETTC) = $ATE + ((OD - (ATE * SPI))/SPI)$
- Or in a simpler form, $ETTC = OD/SPI$

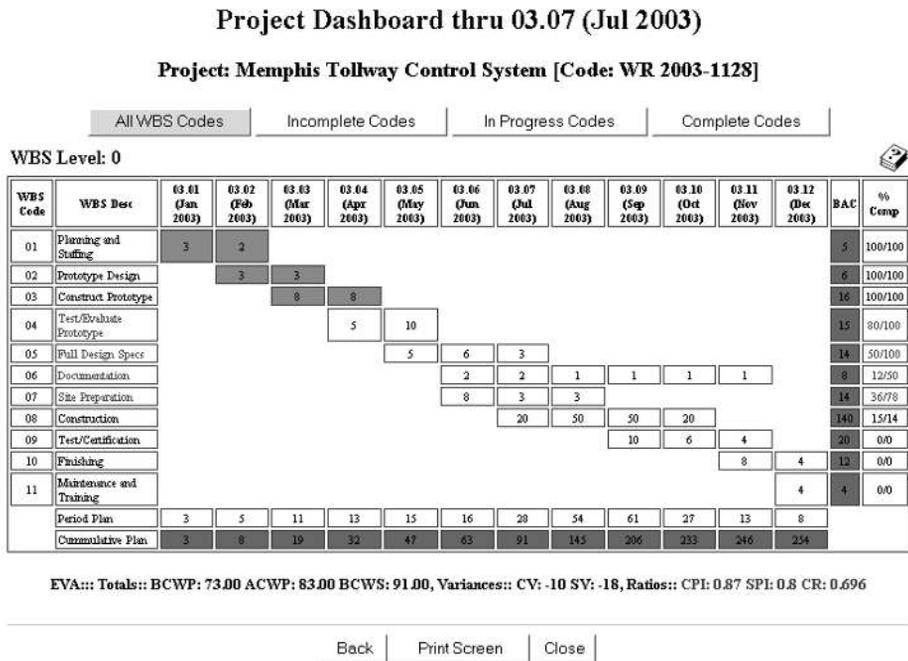
Where ATE is the actual calendar time already expended and OD is the original duration. The SPI is also called the schedule efficiency factor; values less than 1 indicate that the project is taking more time than planned. In the example project herein, the schedule variance is 0.67 months (behind schedule) and the cost variance is \$10,000 (overspent). The estimated time to complete is 15 months instead of the 12 months planned, and the estimated cost to complete is \$289,000 instead of \$254,000. The calculation of these numbers for our example using the previous formulas is implemented in the spreadsheet of Figure 14.13.

Figure 14.14 shows a display from the Web-based FiveAndDime system for this same example showing EVA calculations (as well as percentage complete and cost data) using a “super-charged” Gantt chart format. The color in the Gantt chart bars is green for completed tasks, yellow for tasks in progress, and orange for tasks not yet started. The

Figure 14.13. EV variance calculations in spreadsheet

| Schedule Variances | |
|--|-------|
| Budgeted Cost of Work Scheduled (BCWS) | 91 |
| Budgeted Cost of Work Performed (BCWP, EV) | 73 |
| Schedule Variance (Dollars) | -18 |
| Schedule Variance (Months) | -0.67 |
| Schedule Efficiency Factor | 0.8 |
| Estimated Time to Complete (Months) | 15 |
| Cost Variances | |
| Actual cost of Work Performed (ACWP) | 83 |
| Budgeted Cost of Work Performed (BCWP, EV) | 73 |
| Cost Variance (Dollars) | -10 |
| Cost Variance (Percent) | -3.9 |
| Cost Efficiency Factor | 0.88 |
| Estimated Cost at Completion | 289 |

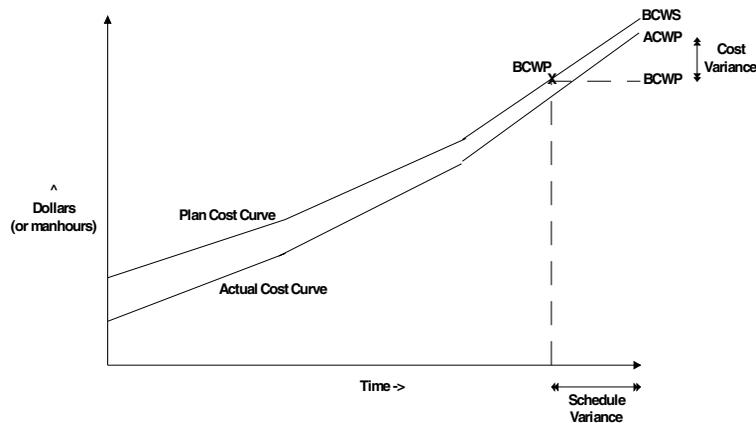
Figure 14.14. EV display in Gantt table



last column for percentage complete shows a ratio where the numerator is the rolled up percentage complete and the denominator is the percentage complete that should have been obtained as of the reporting date; the value of that ratio is actually the SPI for that task. WBS codes that have either the CPI or SPI below unity are displayed in red.

The EV can be projected onto the planned cost (BCWS) curve to graphically show deviations. This is illustrated in Figure 14.15, which shows the planned cost and actual

Figure 14.15. Graphical EV analysis



cost curves for a project analysis, for this example, through July. The schedule variance in time is also shown in that figure along the time axis.

There are other formulas that can be used for ETTC, and some may be better as a project draws closer to completion. In addition, there are EAC formulas that use both the CPI and SPI, and a popular one that is often called the “pessimistic” or “most likely” formula is (Webb, 2003):

$$EAC = ACWP + (BAC - BCWP) / (SPI * CPI)$$

If the SPI is more than 1, then this formula would be less appropriate. The critical ratio (CR) is the product of SPI times CPI; it is also called the cost-schedule index. The CR is the single most important overall metric for time and cost performance. Plotting CR (and also CPI and SPI) versus time is an excellent way to view overall project cost/time health over time and spot trends. Some project management software systems use a dashboard or traffic light indicator for project overall health, and the green-orange-red status color is a function of the project’s CR.

The earned value metrics of CPI, SPI, CR, and EAC are very appropriate and useful in the great majority of projects. However ETTC is problematic and less reliable and should only be used as a very rough estimate (Webb, 2003). Whenever the overall project or any critical task activities go beyond their planned time, then the formula for ETTC goes negative. One adjustment is to modify the previous ETTC formula if the project is already past the original duration (OD):

$$ETTC = OD / SPI$$

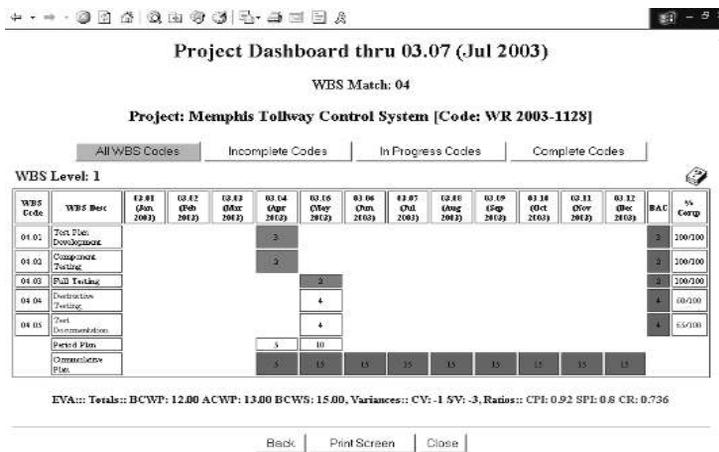
However, a more detailed analysis would yield better results. In a more detail analysis, all incomplete tasks would be reestimated using one of the two ETTC formulas depending upon whether the original duration had been exceeded or not. Then a new critical path would be determined; then from that new critical path a new ETTC found.

EVA greatly facilitates management by exception. When there is a problem with a project as indicated by CPI and/or SPI less than unity, one can drill down in the earned value analysis to lower levels of the WBS. For example, if the overall project index(s) is below unity then one can view the Level 0 WBS codes that have their indexes below unity and then so on to the next level (Level 1 for our example). Some project management software supports this drill down capability and some software systems allow a PM to drill down interactively. Figure 14.16 is a screen shot from FiveAndDime showing a drill down from Level 0 to Level 1. This was accomplished by clicking on a particular WBS code (evaluate prototype) on the Level 0 display (Figure 14.14, where WBS codes with a CPI or SPI below unity are shown in red).

Effective Implementation of EVA

If earned value analysis is so useful, then why is not used more in industry and in particular for IT? There are several reasons for this. First, commercial awareness of earned value is still low; corporate training courses rarely discuss earned value, and there is still relatively little in commercial print on the subject (Brandon, 1999). Even recent comprehensive books on software development project management do not include earned value (Hallows, 1998; Yeates & Cadle, 1996). Second, the data acquisition required (for

Figure 14.16 EV drill down



obtaining percentage complete and actual cost numbers), if implemented in a by-the-book method, may be too costly and time consuming. Third, earned value reporting has not been handled in a usable manner by most project management software. And last, there can be significant employee and contractor resistance problems when trying to put EVA into practice.

Timely and reasonably accurate progress reporting is necessary for EVA, but this data-acquisition process must be easy and inexpensive. Providing for easy and effective data acquisition requires that the methods used are nonintrusive yet provide for the necessary accuracy. Nonintrusive means that very little extra effort is required by project team members or managers to provide the input data for the analysis. The goal is to implement the necessary data acquisition as a by-product of existing corporate reporting or payroll (“time and attendance”) procedures where possible. The key issues in the effective implementation of EVA are (Brandon, 1999):

- Setting appropriate work-packet size
- Progress reporting (defining and obtaining percentage complete information)
- Actual cost basis
- Level of effort tasks
- Human resource issues

If work packets are too small, then project team members will be required to acquire and report excessive data. If work packets are too big, then performance will not be measured frequently enough to allow effective corrective action. If project performance is to be measured monthly (the most common interval), then the work packets should be smaller than a month, such as a week or two. It is a best to tie the maximum size of work packets to the progress and effort data acquisition frequency, especially where employee labor is a significant cost item, as is the typical case with IT projects. In an earlier chapter, WBS packet size was discussed; a recap of the criteria for the packet size is:

- Are realistically and confidently estimable
- Cannot be (or should not be) logically subdivided
- Are to be assigned to one OBS/CBS code
- Can be completed relatively quickly (80 hours maximum rule of thumb; I prefers a maximum of 40 hours)
- Have a meaningful and measurable conclusion and deliverable
- Have enough “visibility” so that progress and cost can be determined
- Can be completed without interruption (i.e., need for more info)

As stated above, each work packet should be assigned to one organization which is responsible for the work to be completed. Additionally if possible, it is also best to have each work packet assigned to a single resource (RBS code), although this may not be

possible in some IT organizations where techniques such as “pair programming” are used. Different types of work should not be lumped into a work package; for example design and coding should be in separate packages, particularly for estimation purposes. Earlier in this book, project calendars and time periods were discussed; time periods in a project calendars should correspond to the interval that is to be used for gathering project performance data, including hours expended and percentage complete estimates. It is important that a work packet not encompass many such periods; the optimal situation would be for a work packet to correspond to exactly one time period. For performance analysis, progress estimates (percentage complete) and cost data should be available at each time period for each work packet (incurred effort and cost during that period). Work packets that are highly interdependent should be combined (if such is possible without extending the time of that packet beyond a few time periods) to reduce overall work packet interdependencies for scheduling.

Progress Reporting

For EVA, percentage complete estimates must be provided for each work packet on a regular reporting basis. For most work packets, taking the time to calculate the percentage complete based on the amount of completed versus the amount of work remaining (or the number of things completed versus the number of things in the work packet) is too time consuming and an unnecessary burden on project team members. If the work packet size has been appropriately determined, then the packet percent estimates can be very rough without losing much accuracy on the overall project performance evaluation.

There are several methods that have been suggested for estimating project percentage complete that provide sufficient accuracy without much additional effort (Christiansen, 1995). The easiest percent completion method is the “all or nothing rule”:

- Work on packet has not started 0%
- Packet is finished 100%

Another a very easy percentage complete scheme is the *halfling*, or 50/50 method:

- Work on packet has not started 0%
- Working on packet has started 50%
- Packet is finished 100%

Another common method is based upon the 80/20 rule (from the notion that 80% of the problems are due to 20% of the work items):

- Work on packet has not started 0%
- Working on packet has started 20%
- Packet is finished 100%

In contracting situations or other situations in which payments may be based upon percentage-complete progress reporting, the contractor can be expected to arrange his or her work to maximize payments. For example, in the 50/50 method, the contractor would start as many work packages as possible; in a 0/100 method, the contractor would break work packets down to a very low level to complete as many packets as possible (Anbari, 2003).

Another method is the weighted milestone, or interim milestone, method in which a percentage complete is allocated for each of several milestones. In most software development methodologies, there is a review, verification, and/or testing step involved in each phase of the methodology (requirements, design, coding, etc.). Common completion and/or monitoring methods include self-checking via unit testing, integration testing, team leader reviews, peer reviews, walkthroughs, or even Fagan inspections (Yeates & Cadle, 1996). Usually, by proper decomposition, that verification step can be subdivided and a portion placed in each work packet. For example, in the implementation or coding phase, the verification step could involve unit testing and/or peer code review (even in pure cleanroom approaches, programmers are going to do some unit testing). With that being the case, the percentage complete can be obtained by a standard interim milestone method where the milestones are:

- Not started 0%
- Started 50%
- Finished packet 75%
- Packet verified 100%

For coding packets, additional milestones can be introduced as:

- Not started 0%
- Started 25%
- Coding complete 75%
- Unit testing complete 90%
- Integration testing complete 100%

Rothman (2004) calls these interim milestones pebbles and prefers the pebbles to be of about a day or two duration. An example of *pebbles* from that reference for software development is:

- Verify design
- Check out code; implement changes
- Develop unit tests
- Successfully compile
- Review code with peer; check code back in
- Develop documentation
- Verification and validation

To be fully useful, any project management performance measurement system must be accepted by project team members and project managers. Thus it is very important that obtaining regular progress (percentage complete) and effort (labor hours) information be easily obtained. The effort required by team members to report their status must be insignificant relevant to the overall work being done. The easiest way to obtain percentage complete and effort statistics is to include such information with existing regular reporting. Some organizations may have timecards and/or progress reports that are completed regularly. The proper time interval for most software projects would be weekly or biweekly; any more frequently would be bothersome, and any less frequently would allow problems to grow before corrective action could be applied. The decomposition of overall work into work packets should be done so that most packets are completed in one or two reporting periods. Whether time cards or progress reports are used, the data obtained must include:

- Work packet ID
- Percentage complete
- Hours expended

If time cards are used, all of an individual's time (paid time) should be accounted for, including project objective work, project level of effort work, and nonproject time (including time off). By doing so, one system can perhaps be used for time and attendance, payroll, and EVA. Figure 14.17 shows an example of a very simple example of such a weekly timecard. Uncompensated overtime (i.e., extra unpaid time put in by

Figure 14.17. Time card

| Employee ID: _____ | | | | |
|--------------------|----------|-------|------------|-------|
| Week #: _____ | | | | |
| Project Code | WBS Code | Hours | % Complete | Notes |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| Total: | | | | |

Copyright © 2006, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.

professional employees) should not be included in EVA analysis, whether the EVA is based on currency or on person hours. If an organization wants to keep track of unpaid time, then separate codes need to be used so that these numbers are not included in EVA analysis (or in payroll calculations). Modern software engineering methodologies discourage such uncompensated time, at least on a regular continuing basis. Some organizations keep an informal log of such unpaid time, and then provide some comp time to employees at the end of the project.

Ease of use and validity of user input are essential for project progress entry. Figure 14.18 shows a screen from the FiveAndDime system for the entry of progress. The resource ID can optionally be automatically obtained from the user's Web log-on. Look-up buttons are provided to allow the user to select resource ID, time period, and WBS code from the database via look-up windows. A look-up window is shown for selecting the WBS code, and if a partial WBS is first entered, then the look-up window will only display the WBS codes that begin with that text. All fields on the form are fully validated. The screen shows the default of five line items; however, when the window opens the user can specify any number of line items. FiveAndDime uses the open source MySQL database, and an organization can also build an SQL (or XML) interface directly into the system to automatically transfer progress information from its own time/payroll system.

Actual Cost Data

As described earlier, EVA uses a percentage complete for each work packet to calculate an earned value for each packet. This value requires knowledge of the planned cost and the percentage complete; it does not require knowledge of the actual cost. *Thus, EVA schedule variances can be calculated without any knowledge of actual costs.* For some projects, the determination of actual costs may not be practical in an accurate and/or timely manner. This may happen when much work is outsourced or when purchased components represent major parts of the system being built or integrated. It can also occur when there is no mechanism to gather costs at the project level; some organizations

Figure 14.18. Progress entry per resource

only code cost to general ledger accounts and have no “job/project costing” system. However, EVA can still be performed partially without any actual cost data including the SPI and estimated time to complete, as well as “drill down” to find problem spots in the WBS in regard to progress.

The total EV cost variances for the project only require knowledge of the total project actual cost not individual work packet actual costs. Thus the CPI and EAC can be determined; however, drilling down to find problem spots in the WBS in regard to cost overruns will not be possible. A major problem in many organizations is that actual costs, while coded to a project, are not subdivided further to the individual WBS codes.

Another problem in many organizations is that while actual costs may be captured at the project and even the WBS level, the close out time of the general ledger and/or job costing system is so slow that it could be many weeks after the end of the month before an actual cost breakdown is available. And while late is better than not at all, it may be too late to take corrective actions. Figure 14.19 illustrates the common method of obtaining actual

Figure 14.19. Actual cost from general ledger

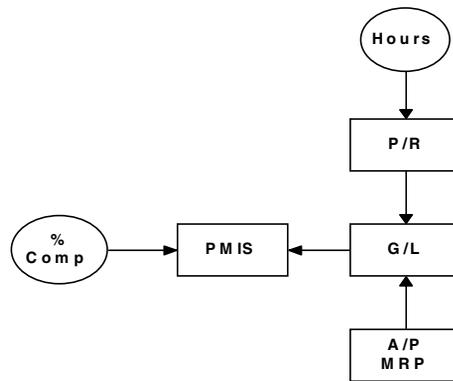
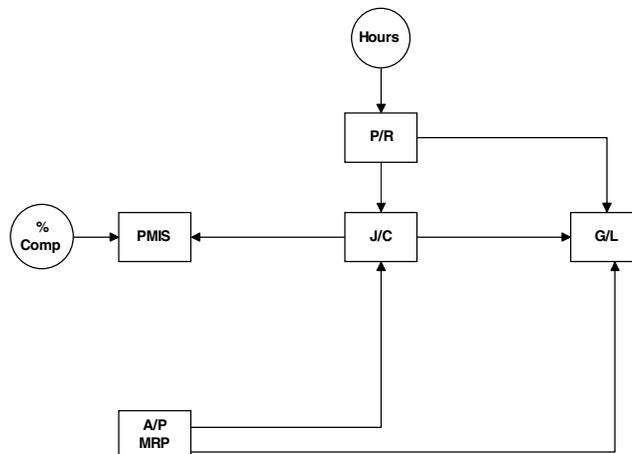


Figure 14.20. Actual cost from job control



cost data from the organization’s general ledger (GL) system into the project management system (PMIS). Usually the actual cost data gets into the GL from payroll (PR), from accounts payable (AP), and/or manufacturing/materials resource/requirements planning (MRP). There are so many major IT systems involved, one can see why there may be significant delays and also significant mispostings and other errors.

Figure 14.20 illustrates the common method of obtaining actual cost data from the organization’s job-costing system (JC). The JC system, however, gets information from PR, GL, AP, and MRP. One can see why there may be significant delays, mispostings, and other errors.

Figure 14.21 illustrates a common hybrid method of obtaining actual cost data from the organization’s IT systems where multiple systems feed actual cost information into the PMIS. This type of hybrid system is an attempt to get cost data into the PMIS faster; however, due to the complication thereof, there may be significant mispostings and other errors. This method may also suffer from the fact that there may be journal entry postings to affected accounts that are need to correct prior period errors (in AP, PR, or MRP), and this results in an out of balance condition between the GL and the PMIS.

In IT most of the cost is usually labor, and that being the case there are several ways that one can expedite the gathering of actual cost information. For IT projects, *one effective EVA method is to represent cost in terms of person hours instead of currency (Brandon, 1998)*. All of the EVA formulas still apply there is just a change to the dimension. This often provides sufficient accuracy for project control purposes, even where there is considerable variation between pay rates on the project team. Even nonlabor costs can be converted back into equivalent person hours using a standard average overall labor rate. This has another advantage for the PM, since it focuses his attention on the matters that he can control.

Another very effective EVA method for IT projects is to set up a feed forward cost reporting system instead of the “feedback” cost system coming from GL (Brandon,

Figure 14.21. Hybrid cost transfer

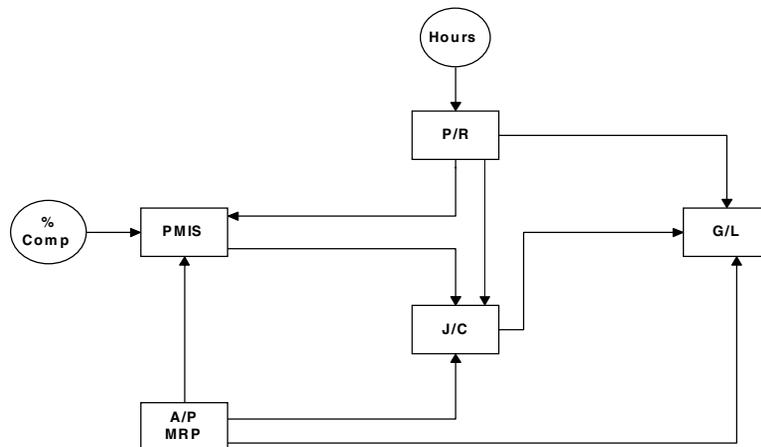
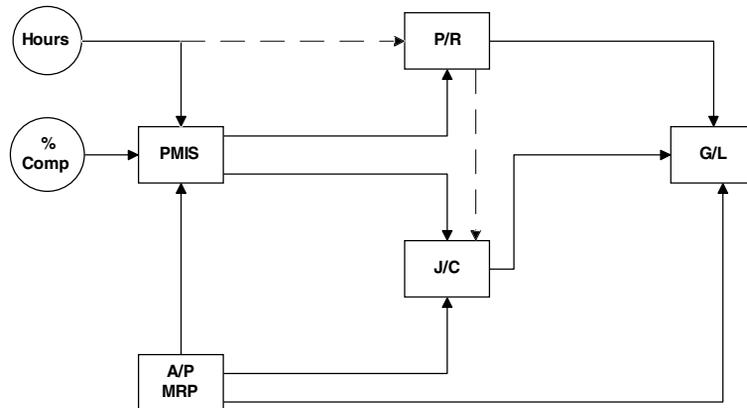


Figure 14.22. Feed forward cost transfer



1998). This is illustrated in Figure 14.22, the hours worked are entered directly into the PMIS, and could also be sent to the PR system directly or indirectly.

With a feed-forward system, resource utilization is entered (as well as percentage complete), and the predicted cost is the resource utilized times the estimated resource rate. For labor, the resource utilization is in person hours. Using these two data items in each reporting period, earned value can be calculated in one of three manners:

- Expressing earned value in terms of person hours (or person days)
- Using the actual pay rate of project workers
- Using categories of skills and skill levels

As discussed in the first option, the entire work plan is expressed in terms of person hours instead of currency. This is the least accurate method, but the simplest.

In the second method, a table of burdened individual pay rates (pay rates multiplied by a factor, which includes all overhead, benefits, etc.) is maintained. This is the most accurate of the three methods, but requires more data to be stored and maintained. There may also be a confidentiality issue in using people's actual pay rates. When this method is used, a computer interface is typically set up between the PR/HR system (which is the system of record for the employee pay rates) and the PMIS.

The third method is fairly accurate yet simple; and it does not pose confidentiality problems. Categories are set up for skills (i.e., analysts, programmers, etc.) and for skill levels (i.e., programmer 1, programmer 2, etc.). For each category an average burdened rate is established (currency per person hour). Then each project worker is assigned one specific category. A spreadsheet or simple database program can be used for these tables also, and linked to the spreadsheet tables previously shown in the EVA; or the PMIS may be able to accommodate this type of information directly, as shown in the screen from FiveAndDime, in Figure 14.23.

Figure 14.23. Table of resources

Page:1

| Resource ID | Project ID | Name | Reference Number | E-Mail | Rate (\$/hr) | Org Code |
|-------------|------------|-----------------|------------------|--------|--------------|----------|
| 1 | 0 | Manager | -- | -- | 50.00 | -- |
| 2 | 0 | Design Engineer | -- | -- | 45.00 | -- |
| 3 | 0 | Fabricator | -- | -- | 40.00 | -- |
| 4 | 0 | Installer | -- | -- | 35.00 | -- |
| 5 | 0 | Task Writer | -- | -- | 30.00 | -- |
| 6 | 0 | Trader | -- | -- | 20.00 | -- |
| 7 | 0 | Trainer | -- | -- | 15.00 | -- |

Current Match: Name Set Name Match Set Ref_Num Match Set Org_Code Match Set Project_ID Match

Add New Resource Print Screen Close

Other EVA Issues

Reporting earned value in many project scheduling software systems (PMIS) is usually not easy for several reasons. First, getting actual costs onto a project so that it can be used in the system's earned value mechanism usually means getting actual costs onto each task, and in many systems getting actual usage onto the resources for each task. Second, setting up an automatic interface between the corporate systems (or departmental databases) and a project management systems is not trivial and usually does not use a standard format such as XML. Third, using the earned value mechanism in these systems is typically not straightforward and simple (Brandon, 1998), and features such as "drill-down" are not available or available only in a batch (noninteractive) manner.

As stated earlier, to be fully successful any project management performance measurement system must be accepted by project team members and project managers. The system must be nonintrusive and *directed primarily towards measuring projects, not measuring individuals*. Because earned value analysis provides a level of insight deep into the project workings, it is likely to be viewed by some as a means of "employee (and/or contractor) evaluation" instead of project performance evaluation. The best way to address this issue is to adopt earned value methodology along with some form of total quality management (TQM). If TQM (or a similar quality effort) is already in place, then earned value could be sealed within that overall envelope. Various quality programs were discussed in the chapter on quality management. Management must focus on the use of earned value for:

- Extrapolation of project cost to complete (EAC)
- Estimation of actual completion date (ETC)

- Identification of project trouble areas (so corrective action can be taken)
- Refinement of work packet estimation methods

and not on the use of earned value for individual evaluations (Brandon, 1998).

For IT projects, costs other than labor may be present and these may or may not be allocated to the project. Contractor and consultant costs are usually part of the WBS and RBS structures. Other direct costs, such as hardware, software, training, entertainment, and travel, may made be part of the WBS or just added to the cost plan developed via the WBS. Indirect, overhead or level of effort (LOE) types of cost may also be assigned to a project and become part of the project budget. These may be general and administrative (G&A), Internal operations and maintenance (I&O), allocations (space, utilities, consumables, communications, etc.), and nonproject time (general meetings, general training, etc.). In some organizations, some of these costs are included in the “burdened rates” for the labor categories. In some organizations these costs are given separate WBS codes. In projectized organizations, and even in traditional line management organizations, as more work becomes team- and project-based, it becomes desirable to associate these costs with projects. If given separate WBS codes, then their contribution to the cost plan is typically a linear, spread over the project life. This is easily handled by software systems in which costs are planned using a combination of time periods and WBS codes. For other software systems in which cost is estimated only at the WBS level for the entire project, handling these overhead costs and percentage complete progress information becomes difficult. One could also exclude these costs from the performance management baseline; however, in a contracted situation this may not be possible. Figure 14.24 shows an example of the input screen from FiveAndDime, where a WBS may be designated as an LOE code, and thus the percentage complete never needs to be input

Figure 14.24. Form to add WBS code (showing LOE and PB)

The screenshot shows a web browser window titled "Add New Entry to Database - Microsoft Internet Explorer". The main heading is "Add New WBS Code". Below this, the project name is displayed as "Project: Memphis Tollway Control System [Code: WR 2003-1128]". The form contains the following fields and controls:

- WBS Code: Text input field.
- Description: Text input field.
- Code Type: Dropdown menu with "Control" selected.
- Master WBS Code: Text input field with a "Lookup" button to its right.
- Performing Org Code: Text input field with a "Lookup" button to its right.
- WBS Risk Factor: Text input field containing the value "1".
- Change Order: Dropdown menu with "No" selected.
- Change Order Reference: Text input field.
- Level of Effort: Dropdown menu with "No" selected.
- Outside PB: Dropdown menu with "No" selected.
- Buttons: "Submit" and "Reset" buttons at the bottom.

(the EV will equal the planned cost through the current period yielding no schedule variation). Also as seen in that same screen shot, a WBS may be designated as outside of the performance baseline (PB) in which case it will not be included in the EVA analysis. WBS codes may also be designated as baseline or change order, and EVA can be done with or without change orders.

As well as the web based FiveAndDime system, there are a number of other software products (both mainframe and client-server based) that were specifically made to handle earned value in a comprehensive manner including:

- C-Brat (www.cbratproject.com)
- PMOffice (www.systemcorp.com)
- Cobra (www.welcom.com)
- Microframe's MPM (www.businessengine.com)
- C/S Solutions's WinInsight (www.cs-solutions.com)
- DeckKer's Tracker (www.dtrakker.com)

EVA, Success Factors, and Stage Gates

Stage gates are the traditional way that project performance is monitored and controlled by upper management of both the benefiting and performing organizations. These stage gates may be event driven by project milestones or they may be periodic. The most common scenario for IT projects is periodic (i.e., monthly) using the traditional Gantt chart and budget-versus-actual cost data (as was discussed earlier in this chapter) plus whatever verification (product testing) and/or validation (user satisfaction) information is available at those times. Figure 14.25 illustrates this traditional stage gate approach.

The problem with this traditional approach is that considerable time (and cost) is expended at the stage gate review meetings, because the PM, users, line management of both the performing and benefiting organizations, and some project team members are

Figure 14.25. Traditional stage gate

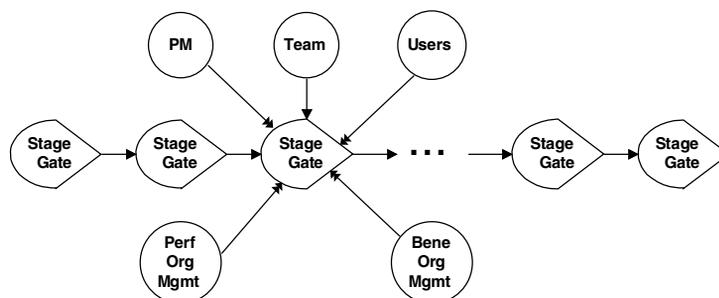
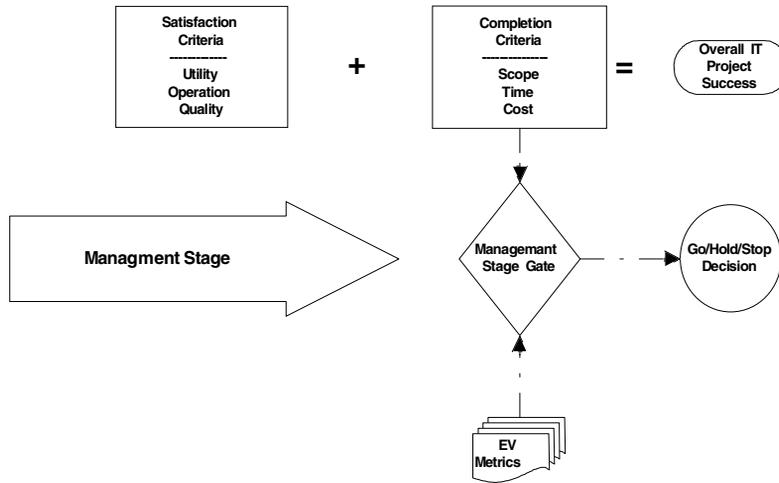


Figure 14.26. Management stage gate (using EVA)



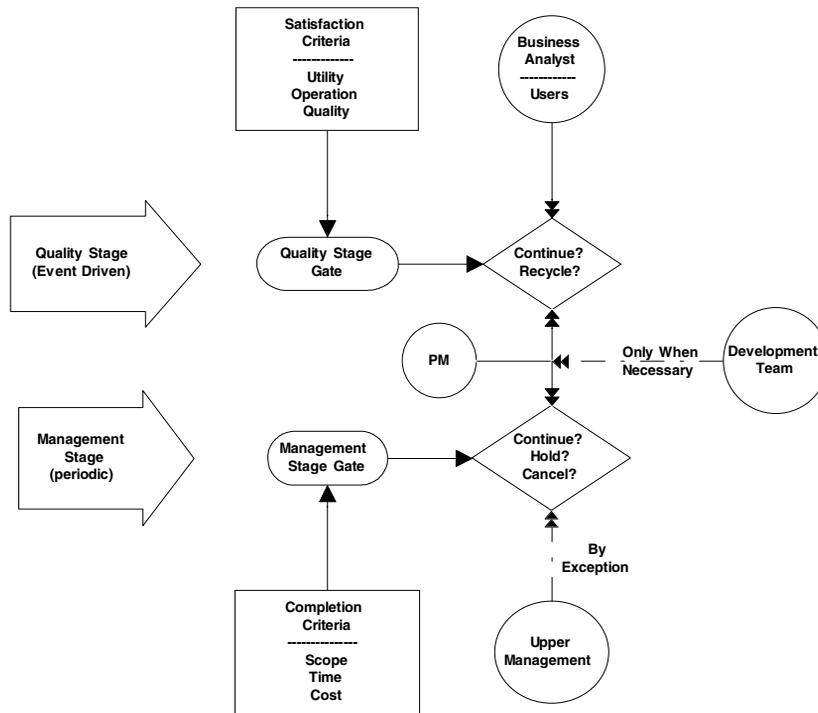
generally present. The project team members typically in attendance include the business analysts and, in some cases, even key designers and developers.

If the overall stage gate is divided into completion and satisfaction reviews, then both the frequency and number of participants can be reduced resulting in the consumption of less project time and cost. In earlier chapters, project completion and satisfaction success factors were discussed, and the notion of separating the gating process into these categories was presented. This dual gating process was illustrated in Figure 2.6 in Chapter II, where management stage gates are placed at regular time intervals (i.e., monthly or quarterly), and quality stage gates are placed at key delivery points for product manifestations (i.e., requirements documentation, use cases, paper prototypes, working prototypes, test results, etc.). There may be multiple quality stage gates within a management stage gate time interval, or vice versa.

EVA is used in the management stage gate not only to evaluate the completion factors, but the EAC is also used to reevaluate the chosen project benefit metric(s) as shown in Figure 14.26. The chosen benefit metrics might be any of those discussed earlier in the book such as IRR or scoring methods. The estimated completion date might also be used in recalculating benefit metrics, if calendar completion time is a key factor in the benefit analysis. It is important that these management reviews do not use sunk cost (how much has already been invested/spent) in determining whether a project should proceed or not; the only cost considered should be the estimated cost at completion versus the benefit. Keeping a badly performing project alive by sunk cost is a trap into which many organizations have fallen.

The dual gating approach not only provides for the exposure and analysis of both project completion and satisfaction criteria, but also minimizes the time of key individuals in meetings. This is illustrated in Figure 14.27, where upper management and the develop-

Figure 14.27. Dual stage gates and stakeholder involvement



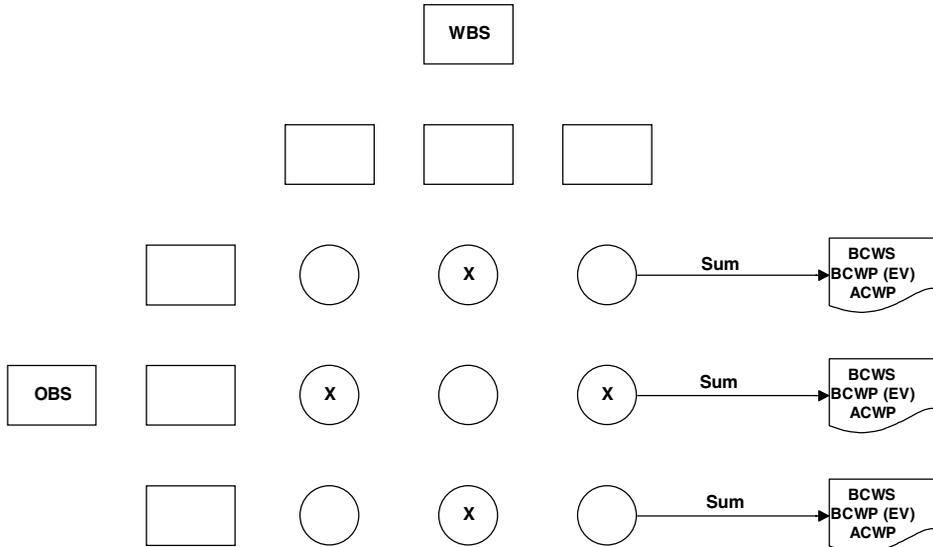
ment team are relatively isolated from regular meetings. The management stage gate reviews, while periodic, need to be held only if the EVA values are not good (CPI and/or SPI below unity). In addition, upper management does not need to be involved in the quality stage gates at all, and neither does most of the project team; users, business analysts, and the PM may be the only personnel needed for these gate reviews.

Since earned value metrics are currency based, both schedule variance and cost variances can be rolled up along organizational lines for one project or even for multiple projects. In this way, the overall performance of organizational components can be evaluated. This is illustrated in Figure 14.28. A decision is made in regard to the management level for the organization, and the earned value metrics assigned to those organizational components are summed up. This is discussed further in Chapter XVI.

Chapter Summary

In this chapter earned value analysis has been discussed and illustrated in detail. EVA is often difficult to implement effectively and can have a number of problem areas.

Figure 14.28. EVA rollup by organizational component



However, this chapter has identified the problem areas and their practical solutions. EVA is the only practical method to determine actual project status and develop reliable estimates for completion cost and time. EVA should be included in all IT projects of any significant size. EVA is one of the key metrics in the management-for-success philosophy that has been developed in this book via critical success factors and dual stage gates.

References

- Abba, W. (2000). *Earned value management rediscovered*. Retrieved from http://www.acq.osd.mil/pm/newpolicy/misc/abba_art.html
- Anbari, F. (2003, December). Earned value project management method and extensions. *Project Management Journal*, 34(4).
- Brandon, D. (1998, June). Implementing earned value easily and effectively. *Project Management Journal*, 29(2).
- Brandon, D. (1999). Implementing earned value easily and effectively. In J. Pinto (Ed.), *Essentials of Project Control* (pp. 113-124). Newton Square, PA: Project Management Institute Press.
- Christensen, D., & Ferens, D. (1995). Using earned value for performance measurement on software projects. *Acquisition Review Quarterly*, Spring, 155-171

- Christensen, D., et al. (1995). A review of estimate at completion research. *Journal of Cost Analysis*, Spring.
- Fleming, Q., & Koppelman, J. (1996). *Earned value project management*. Project Management Institute.
- Fleming, Q., & Koppelman, J. (1998). *Earned value project management*. A Powerful Tool for Software Projects. *CROSSTALK, The Journal of Defense Software Engineering*, July, 19-23
- Hallows, J. (1998). *Information systems project management*. New York: Amacom.
- Kiewel, B. (1998). *Measuring progress in software development*. PMNetwork, January, 29-32
- Lewis, J. (1995). *Project planning, scheduling & control*. In Irwin.
- Rothman, R. (2004, February 16). Using pebbles to track project state. *Computerworld*.
- Webb, A. (2003). *Using earned value*. Burlington, VT: Gower.
- Yeates, D., & Cadle, J. (1996). *Project management for information systems*. London: Pitman.