Automatic Creation of Countermeasure Plan against Process Delay: Creation of Countermeasures based on Crashing and Fast-tracking

Yuichi INOUE †, Kazuki MORI †, and Seiichi KOMIYA †
† Graduate School of Engineering and Science, Shibaura Institute of Technology
3-9-14 Shibaura, Minato-ku, Tokyo, 108-8548
JAPAN
nb11101@shibaura-it.ac.jp

Abstract: - Success or failure of a software project mainly depends on the management capability of the project manager. Therefore, EVA (Earned Value Analysis) is gaining much attention recently as a method of managing a project in an integrated fashion by introducing a unified metric called EV (Earned Value) and quantitatively measuring and analyzing the cost and schedule of a project. Project managers can use EVA to estimate SEAC (Schedule Estimate At Completion) and EAC (Estimate At Completion). However, since EVA does not include constraints as a factor for assigning human resources, such as the available periods of each person, etc., EVA quite often generates inaccurate estimations for SEAC and EAC. In addition, suppose that a situation arises due to a delay in project progress, in which the deadline cannot be met if the project proceeds as is without countermeasures adopted. In this case, it is necessary to provide a prospect that indicates whether the project can be completed within the original delivery deadline (i.e., the original delivery deadline can be met) or not if any countermeasures are adopted to recover the process delay. If there is some way to bring the project back to its original schedule and complete it within the assigned time table, it is necessary to be able to present the countermeasure or an actual development plan. However, EVA cannot be used to show such a prospect or present the means to restore the project to its original timeline even if such countermeasures exist. This paper proposes a method to solve these problems and discusses its effectiveness by comparing the authors’ approach with that of the EVA.

Key-Words: - Schedule Planning for Software Development, Countermeasure Plan against Process Delay, Crashing, Fast-tracking

1 Introduction

Success or failure of a software project mainly depends on the management capability of the project manager. For this reason, PMBOK (Project Management Body of Knowledge) is arranged and Modern Project Management[1] is being made pervasive. PMBOK has employed the management method called EVA (Earned Value Analysis)[2] to make it possible to quantitatively measure and analyze the cost and schedule of a project and manage a project in an integrated fashion by introducing a unified metric called Earned Value. Several methods have been proposed to estimate the future cost and progress based on this approach. However, since EVA does not include constraints such as the available periods of each person, etc. as a factor for assigning human resources, EVA quite often generates inaccurate estimations for SEAC (Schedule Estimate At Completion) and EAC (Estimate At Completion). In addition, to evaluate the project progress, it is not enough to estimate the SEAC (Schedule Estimate At Completion) and EAC (Estimate At Completion). Suppose that a situation arises due to a delay in project progress, in which the deadline cannot be met if the project proceeds as is with no countermeasures adopted to recover the process delay. In this case, it is necessary to provide a prospect that indicates whether the project can be completed within the original delivery deadline (i.e., the original delivery deadline can be met) or not if any countermeasures are adopted to recover the process delay. If there is some way to recover the process delay and complete it within the assigned time table, it is necessary to be able to present the countermeasure or an actual development plan. Therefore, the authors propose a method to estimate SEAC (Schedule Estimate At Completion) and EAC (Estimate At Completion) by taking into account the constraints for assigning human resources and describe the estimation of SEAC and EAC. The authors also propose a method that can automatically generate an alternative development plan which can recover the process delay when it becomes clear that the deadline cannot be met if no countermeasure is adopted to recover the process delay.
2. The Concept of Earned Value (EVA)

2.1 Conceptual diagram of EVA

EVA is one of the methods used to quantitatively manage the progress of a project by measuring the progress and comparing the measurements with the planned values. The following terms are used in EVA:

- **PV**: Planned Value
- **EV**: Earned Value
- **AC**: Actual Cost
- **BAC**: Budget At Completion
- **CV**: Cost Variance
- **CPI**: Cost Performance Index
- **EAC**: Estimate At Completion
- **VAC**: Variance At Completion
- **SAC**: Schedule At Completion (Planned working days until project completion)
- **SV**: Schedule Variance
- **SPI**: Schedule Performance Index
- **SEAC**: Schedule Estimate At Completion
- **SVAC**: Schedule Variance At Completion

The conceptual diagram of EVA in Figure 1 illustrates the concept of EVA. In EVA, every development task is evaluated from the viewpoint of its cost based on the assumption that the development budget is consumed as time advances. Under this assumption, the budget cost of tasks is referred to as PV, which should be achieved before a certain point and has been estimated at the planning phase. Based on this definition, PV can be regarded as a function that represents the consumption plan of the development budget as a function of the elapsed time. The value of PV is a cumulative total value of the development cost planned to consume from the beginning of the project to PT (Present Time). In Figure 1, PV is represented by the dashed line and the intersection point of PV and PT represents the current value of PV. As a special case of PV, BAC represents the total value of PV planned to be consumed until the project is completed (SAC). In Figure 1, the intersection point of PV and SAC represents the value of BAC. On the other hand, EV represents the project progress in terms of consumed cost calculated from the elapsed time of the project. Therefore, in many cases, the current value of EV is equal to the value generated by multiplying PV by the progress rate at the present time (PT). In Figure 1, EV is represented by the dotted and dashed line and the current value of EV is represented by the intersection point of PV and AP (Actual Progress). On the other hand, AC represents the cumulated cost actually spent at the present time (PT). In Figure 1, AC is represented by the solid line and the current value of AC is represented by the intersection point of the AC line and PT. Now use the sample project to explain the concept of EVA.

2.2 Sample project

This Section introduces an example of project (sample project) that discusses the development of a demonstration system for an exhibition.

This project is organized on July 2 and the system is planned to be delivered on July 31. The budget is 1,188,800 yen. The system development tasks include System Architecture design (SA), System Design 1 (SD1), System Design 2 (SD2), Design Review (DR), Detailed Design 1 (DD1), Detailed Design 2 (DD2), Development of Test Support Tool (DT), Performance Evaluation (PE), Coding & Unit Test 1 (C/UT1), Coding & Unit Test 2 (C/UT2), and Integration Test (IT). The following is a set of constraints that should be taken into account when developing a software development plan. Since the demonstration system is required to be a high performance system, it is necessary to achieve the performance expected before the project is launched. Therefore, a performance evaluation is required on a section of the detailed design and the evaluation result must be reflected on the coding tasks. Conducting the performance evaluation requires the performance analytic technology. The hardware used in the exhibition is a newly developed machine that is now under development for presentation. Therefore, it is not available until July 24. Furthermore, it will be transported to the exhibition hall on July 31. This device is essential for the integration test. Since the available period of this machine is under the stringent limitation described above, a test support tool is developed to enable an efficient test. As a general rule, no work is done on Saturday and Sunday since they are holidays. In the sample project, four members A, B, C, and D are assigned as the personnel in charge of design, two members E and F are assigned as the personnel in charge of quality assurance, and two members G and H are assigned as the backup members for design tasks. Table 1 shows the assignment condition for each member. The development plan is made and the project is started assuming that labor hours are eight hours per day. The development plan at the start of the project is listed at the top row in each process in Figure 2.
Figure 2. The original schedule and the status in the case of process delay

Figure 1. A conceptual diagram of EVA

Table 1. Human resource information

<table>
<thead>
<tr>
<th>Worker</th>
<th>Skill</th>
<th>Rank of skill</th>
<th>Available period</th>
<th>Cost per hour (Yen)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>System analysis a</td>
<td></td>
<td>7/25 –</td>
<td>2500</td>
</tr>
<tr>
<td></td>
<td>Design technology b</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Programming c</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>System analysis b</td>
<td></td>
<td>7/25 –</td>
<td>3500</td>
</tr>
<tr>
<td></td>
<td>Design technology a</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Programming a</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>System analysis a</td>
<td></td>
<td>7/18, 8/1 –</td>
<td>2800</td>
</tr>
<tr>
<td></td>
<td>Design technology b</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Programming c</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Design technology b</td>
<td></td>
<td>7/10 – 7/25</td>
<td>3000</td>
</tr>
<tr>
<td></td>
<td>Programming a</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>Quality assurance technology b</td>
<td>7/25 –</td>
<td>3200</td>
<td></td>
</tr>
</tbody>
</table>

After the project is started, by the end of July 11, two days worth of delay has occurred in the progress. As a result, the work for DR has to be completed in four days. The progress status of the project after the process delay is detected is listed in the middle row in each process in Figure 2.

Figure 1 shows the skills of each member and the degree of proficiency of each member’s skill by four levels of \{a: excellent, b: good, c: fair, d: poor\}.

The AC (Actual Cost) spent by the end of July 11 is 150,000 yen for the SA phase, 60,000 yen for the SD1 phase, 91,000 yen for the SD2 phase, and 281,600 yen for the DR phase.

2.3 Evaluation of the project progress status by EVA

In the sample project, the first process delay arose when the DR process completed. The work period of DR was extended to four days from the two days initially scheduled due to the delay, thus the progress rate of this task at this point is 50%. EVA typically estimates SEAC (Schedule Estimate At Completion) and EAC (Estimate At Completion) by evaluating the actual progress and the results of the project when it is completed from 25 to 30%.

However, no process delay is detected at this point (when the DR process is started in this case). As a result, we used EVA to evaluate the project progress status when a process delay is first detected. Table 2 lists the PV value, the progress rate, the EV value, the AC value, and the total values for these terms (the average value for the progress rate).

EVA makes it possible to manage the progress and the cost at the same time by converting the planned and actual values into monetary values.

On the other hand, it is necessary to perform the following three functions to evaluate the project progress status.
Table 2. The analysis result produced by EVA when process delay occurred

<table>
<thead>
<tr>
<th>Process</th>
<th>PV</th>
<th>Progress rate</th>
<th>EV</th>
<th>AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
<td>144,000</td>
<td>100.00%</td>
<td>144,000</td>
<td>150,000</td>
</tr>
<tr>
<td>SD1</td>
<td>60,000</td>
<td>100.00%</td>
<td>60,000</td>
<td>60,000</td>
</tr>
<tr>
<td>SD2</td>
<td>84,000</td>
<td>100.00%</td>
<td>84,000</td>
<td>91,000</td>
</tr>
<tr>
<td>DR</td>
<td>140,800</td>
<td>50.00%</td>
<td>70,400</td>
<td>281,600</td>
</tr>
<tr>
<td>Total</td>
<td>428,800</td>
<td>83.60%</td>
<td>358,400</td>
<td>582,600</td>
</tr>
</tbody>
</table>

(F1): The function to estimate SEAC (Schedule Estimate At Completion) and EAC (Estimate At Completion) if the project proceeds as is with no countermeasures adopted to recover the process delay when a process delay occurs.

(F2): Suppose that the project proves to be impossible to meet the original delivery deadline by applying the function (F1). In this case, the function (F2) determines whether the project can be completed within the original delivery deadline (i.e., the original delivery deadline can be met) or not if any countermeasure is adopted to recover the process delay.

(F3): The function to automatically generate the recovery plan that presents a concrete development plan (development schedule and assignment of personnel) if it is determined that the deadline can be met.

As discussed above, EVA can provide the function (F1). However, EVA cannot provide a prospect that indicates whether the deadline can be met or not, or cannot present a concrete development plan to restore the project to its original timeline when determined that the original deadline can still be met. Therefore, it is not possible to use EVA alone to evaluate the project progress status.

Based on the above discussion, we propose a method that can provide the three functions to evaluate the project progress status based on the constraints.

3. A Proposal of a Method for Evaluating the Project Progress Status Based on the Constraints

3.1 Constraints of Software Development Planning Problems

This section discusses the constraints inherent to the software development planning problems, which plays important roles for evaluating the project progress status. The authors have been engaged in studying and developing the system that can automatically generate software development plan including the development schedule and personnel assignment. In Reference to [3-12], we pointed out that the initial software development plan developed prior to starting the project has to satisfy the following constraints from (C1) to (C4).

(C1) Constraints on operational sequence

Actual operational sequences of the software development processes depend on intermediate products.

Operation of Process b requires that the product of Process a, i.e. Intermediate product α, is created prior to the start of Process b. This condition is referred to as "the precondition of Process b." And, operation of Process c requires that the product of Process b, i.e. Intermediate product β, has been created prior to the start of Process c. This condition is referred to as "the post-condition of Process b." Thus the operational sequence of Processes a, b, and c is determined by the Intermediate products α and β. These constraints are referred to as "the constraints on the operational sequence."

(C2) Constraints imposed by the condition of resource assignment

Each software development task should be assigned with the human resources (personnel) and/or non-human resources (machine environments, etc.) that have necessary skills, qualifications and/or capabilities for that particular task. These constraints are referred to as "the constraints on resource assignment conditions."

(C3) Constraint on the assignment period of resource

Furthermore, each task of software development has another constraint that the qualified resources are available only for their assignable period (i.e. when they are not fully booked and available for such tasks). These constraints are referred to as "the constraints on resource assignable periods."

(C4) Constraints on resource capacity limitation

This paper introduces the concept of "capacity" in order to represent respective resource capacity limitation, and present it as an attribute of the resources. In particular, a resource's capacity is defined by the upper limit value of each resource's working rate (in percentage). That is, the total working hours for a day as assigned to the resource (or, the total working hours for the set of tasks in the event that single resource is assigned to a set of tasks that should be performed in parallel) should be
divided by daily workable hours for such a resource and then the result is multiplied with 100 to derive the working rate. The predefined upper limit for the working rate of each resource is referred to as the constraints on the resource capacity limitation of given resource.

3.3 The constraint-based method to estimate SEAC (Schedule Estimate At Completion) and EAC (Estimate At Completion)

The author employs a method referred to as the perturbation-based repercussion analysis of process delay [11, 12] in which the impact of a delay occurred in a phase on the succeeding processes is analyzed by taking into account the constraints. We proposed the method to perform a perturbation-based repercussion analysis when a process delay occurs in a software development project in Papers [11,12]. It is possible to estimate the SEAC (Schedule Estimate At Completion) by applying the perturbation-based repercussion analysis when a process delay occurs. Chapter 4 illustrates a concrete example of the perturbation-based repercussion analysis.

3.4 The constraint-based method to automatically generate a countermeasure plan

One of the problems of EVA is its lack of means to present an alternative development plan which can complete the project by the original delivery date when a process delay occurs, even if such a plan exists. On the other hand, the constraints-based method, as its greatest strength, can automatically generate an alternative development plan which can recover the process delay and can complete the project by the original delivery date when it is possible (we have studied and developed an automatic generation tool) [8,11].

Typical countermeasures that can recover process delays by modifying the development plan (schedule and personnel assignment plan) are classified into the following three types and their combinations.

A) Crashing[11]

Crashing is a method to shorten a process period by assigning additional excess personnel (not booked yet) to the process, who satisfy the assignment conditions.

B) Fast-tracking[8]

Fast-tracking is a method to shorten a process period by starting the subsequent process before its preceding process has finished in order to execute processes in parallel which were originally planned to be executed sequentially. Since the subsequent process starts before its preceding process has finished, it may result in decreased quality of the intermediate products of the subsequent process. To avoid this issue, it is necessary to assign personnel with excellent skills to the subsequent process.

The methods to generate a countermeasure plan of process delay for (a) and (b) are identical to the one described in Section 3.2 “The mechanism for automatically generating the development plan based on the constraints.” The following discusses the method to generate a countermeasure plan of process delay from the viewpoint of relationship with the constraints that are inherent to planning of software development:

Crashing employs the constrains (C1) – (C4) as they are and assigns additional (excess) personnel who satisfy these constraints. Fast-tracking parallelizes the processes that are planned to be sequentially executed by employing the constraints (C2) – (C4) as they are and loosely applying the constraint (C1) to particular processes. Then, to keep the quality of intermediate products generated in the subsequent process, it applies stricter personnel assignment conditions by adding the constraint (C6). Specifically, the constraint “(C6) the skill level of Worker β must be equal to or higher than that of Worker α” is added as the constraint that must be satisfied by Worker β who is assigned to Process B subsequent to Process A, where Worker α is assigned to the preceding Process A.

4. Estimation of SEAC (Schedule Estimate At Completion) and EAC (Estimate At Completion)

4.1 Estimation of SEAC (Schedule Estimate At Completion) and EAC (Estimate At Completion) by the constrain-based method

This section uses the sample project to illustrate the perturbation-based repercussion analysis method used to calculate the SEAC (Schedule Estimate At Completion) when a process delay occurs.

The first step is to analyze the impact on the process of which prerequisite condition includes the process in which a process delay has occurred. DR has been delayed for two days and it was terminated on July 13. According to the constraint on the
execution sequence of work, delays occur in the phases DD1, DD2, and DT since they have DR as their prerequisite condition. Thus, the work period of DD1 and DD2 is shifted to July 16-18, and DT is shifted to July 16-24.

The impacts of the process delay on the succeeding processes are analyzed in the same way. Since DR finishes on July 13 due to the two days of delay, the phase DD2 following DR is shifted to July 16-18. The work period of PE that is the subsequent phase of DD2 must be shifted to July 19-20 accordingly, however, the member C who is in charge of PE is available for assignment only in the period of July 10-18 and August 1-5 according to the constraints on his/her assignment period. In addition, there is no person other than C who has the skill to perform this work. As a result, PE is shifted to August 1-2. Then, C/UT2 that is the subsequent phase of PE is shifted to August 3-9 and IT that is the subsequent phase of C/UT2 is shifted to August 10-14. As the result of analysis in the above discussion, SEAC has been found to be shifted for two weeks (10 actual working days) from the originally scheduled day (Figure 3).

While the number of actual working days until SEAC calculated by EVA is delayed four days, the number of actual working days calculated by the constraints-based method is extended by ten actual working days. This difference is resulted from whether the constraints (C1)-(C4) pertaining to the problem of software development planning described in Chapter 3 is taken into account or not when estimating the SEAC.

As the result of the perturbation-based repercussion analysis described in Chapter 4, it becomes clear that it is absolutely necessary to start the PE process on July 11 as initially scheduled in order to meet the delivery deadline of July 30 due to the constraint on the assignable period of Worker C. Now we can apply the methods of Crashing, Fast-tracking, Holiday work and their combination as the means to change the development plan (development schedule and personnel assignment) to recover the process delay. Therefore, it is necessary to decide whether one of these methods or any combination of these methods are employed to make it possible to start the PE process on July 17 as initially scheduled.

The authors have already clarified several methods with which the countermeasure is generated by Crashing[11] or Fast-tracking[8] and discussed their implementation methods in several papers. The following illustrates how to check if any countermeasure plan exists with Crashing, Fast-tracking, Holiday works, or their combination to recover the process delay, and the automatically generated countermeasure plan if such a countermeasure exists. Since the existence of countermeasure plan is checked on July 12, only the schedule after July 13 is reflected to the countermeasure plan (the schedule for July 12 and earlier cannot be modified).

5.1 Countermeasure plan by Crashing

We tried automatically generating a countermeasure plan by Crashing to recover the process delay. As a result, more than one countermeasure plan was generated. However, none of these plans could complete the project by the delivery deadline initially set. The reason is as follows:

To complete the project by July 30, it is necessary to start the PE process on July 17 as initially scheduled. Since the DR process that is the preceding process of the DD2 process finishes on July 13, the DD2 process has to be completed in the period from July 14 to 16. However, since there is only one working day, July 16, in this period, it is not possible to complete the DD2 process even if the excessive personnel are assigned. As a result, applying Crashing alone cannot complete the project by July 30.

5.2 Countermeasure plan by Fast-tracking
We tried automatically generating a countermeasure plan by Fast-tracking to recover the process delay. As a result, more than one countermeasure plan was generated. However, none of these plans could complete the project by the delivery deadline initially set. The reason is as follows:

To complete the project by July 30, it is necessary to start the PE process on July 17 as initially scheduled. However, since it is not possible to start the DD2 process by Fast-tracking before July 13, the DD2 process has to be completed in the period from July 13 to 16. Since it takes three days to complete the DD2 process when worked by Worker B alone and there are only two working days, July 13 and 16, in this period, it is not possible to complete the work in two days. As a result, applying Fast-tracking alone cannot complete the project by July 30.

5.3 Countermeasure plan by combination of Crashing and Fast-tracking

It shows that applying Crashing or Fast-tracking alone cannot generate a plan that can complete the project by the deadline initially set. Then, we tried to generate a countermeasure plan by combining Crashing and Fast-tracking. As a result, more than one countermeasure plan was generated. Among these plans, only one plan can be used to complete the project by the delivery deadline initially set. Figure 4 shows this plan. The countermeasure plan in Figure 4 shows that assigning additional Workers G and H by Crashing and adding two normal working days for these personnel on July 13 and 16 by Fast-tracking makes it possible to complete the DD2 process. Therefore, combination of Crashing and Fast-tracking makes it possible to complete all of the processes by July 30. It is proved that the project can be completed on the initial deadline day.

In addition, creating a table similar to Table 3 indicates that EAC can be calculated as 1,360,800 yen. As a result, the plan generated by Holiday works is selected as the countermeasure plan for process delay by comparing the cost of this plan with the cost of the plan by Holiday work.

### Figure 4. A countermeasure plan for process delay generated by combination of Crashing and Fast-tracking

<table>
<thead>
<tr>
<th>No.</th>
<th>Process</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>44,800</td>
</tr>
<tr>
<td>2</td>
<td>C/UT1</td>
<td>100,000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>100,000</td>
</tr>
<tr>
<td>3</td>
<td>C/UT2</td>
<td>140,000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>140,000</td>
</tr>
<tr>
<td>4</td>
<td>IT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>76,800</td>
</tr>
<tr>
<td>5</td>
<td>IT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>86,400</td>
</tr>
<tr>
<td>6</td>
<td>IT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>163,200</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>360,000</td>
<td>504,000</td>
<td>134,400</td>
<td>168,000</td>
<td>76,800</td>
<td>86,400</td>
<td>1,329,600</td>
</tr>
</tbody>
</table>

### Table 3. The table used to calculate EAC by the authors’ method when a process delay occurs

6. Comparison with Relevant Studies

There are many different models that attempt to present typical work structures of software development, including PMDB [14], Design-Net [15, 16], kyotoDB [17], and PROMX [18]. However, these models do not explicitly address either the relationship between software development tasks and the resources essential to conduct the tasks or the constraints on the conditions and available periods of resource assignment, although they are useful as models to represent the task model of a project because they focus on how to represent the hierarchy and sequence of tasks (Although PMDB uses Person as an entity, it does not address the constraints related to the resource allocation conditions and the available period of resources). Therefore, they are
not adequate for the project management models of software development projects.

We also mentioned CCPM (Critical Chain Project Management) [13, 19, 20, 21] getting attention recently in comparison with our approach. Our approach is different from CCPM in the following points.

1. Our viewpoint of man-hours estimation is different from that of CCPM. CCPM adopts man-hours estimation of each process with 50% of success probability and uses the joining and project buffers to reduce the risk of process delay due to estimation errors. In our approach, an average of extra man-hours is calculated for the joining and project buffers and assigned to each process.

2. Our viewpoint of progress management is different from that of CCPM. CCPM manages the progress of the whole project by examining the consumption ratio of the buffer instead of managing the progress of each process. For this reason, CCPM can be used to detect process delays of the whole project, but it is not adequate for understanding the progress of processes which are not on the Critical Chain. In our approach, progress is managed by each process. As a result, it is possible to understand the progress of every process, regardless of if it is on the Critical Path.

3. When a process delay is detected, it is not easy to change the project schedule in CCPM to recover the delay, but in our approach as described in this paper and Reference [8, 11], it is possible to use our tool to develop a revised plan dynamically that can be used to recover the process delay.

7. Conclusion

In this paper, we described the concept of EVA and introduced a concrete example of project (referred to as a sample project) that used EVA to demonstrate the future cost and progress. Then, the authors discussed the SEAC (Schedule Estimate At Completion) and EAC (Estimate At Completion).

We discussed that we can apply two methods, (a) Crashing, (b) Fast-tracking, and their combination as the means to change the development plan (development schedule and personnel assignment) to recover the process delay. Then, we applied the constraints-based method to the sample project and tried to automatically generate a countermeasure plan. As a result, we showed that it is possible to automatically generate a concrete countermeasure plan that completes the project by the initially set delivery deadline by the combination of Crashing and Fast-tracking methods. The results of the constraints-based method indicated that the functions (F2) and (F3) can be achieved for evaluation of the project progress status.

We proposed the constraints-based method to evaluate the project progress status and discussed the effectiveness of the method.

References:


