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

Yuichi INOUE†, Kazuki MORI†, and Seiichi KOMIYA†

† Graduate School of Engineering and Science, Shibaura Institute of Technology

3-9-14 Shibaura, Minato-ku, Tokyo, Japan 108-8548

nb11101@shibaura-it.ac.jp

***Abstract*—** A lot of whether or not a software project is successful depends on the management capability of the project manager. Therefore, EVA (Earned Value Analysis) has received a lot of attention recently as a method for 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 is a method without taking account of the constraints on assigning human resources: the available periods of each person, and so on. EVA quite often generates inaccurate estimations for SEAC and EAC. In addition, suppose that it is not possible to meet the deadline due to process delay if the project proceeds without taking any measures to recover the process delay. In this case, it is necessary to provide a prospect that indicates whether or not the project can be completed within the original delivery deadline (i.e., it is possible to meet the deadline) if any countermeasures are adopted to recover the process delay. If there is some way to return the project schedule 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 show any countermeasures which it is possible to restore the project schedule to its original schedule 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.

***Keywords*—** Schedule Planning for Software Development, Countermeasure Plan against Process Delay, GA, Crashing, Fast-tracking, Holyday-working.

I. Introduction

A lot of whether or not a software project is successful 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 growing in popularity. 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 is a method without taking account of the constraints on assigning human resources: the available periods of each person, and so on. 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 it is not possible to meet the deadline due to process delay if the project proceeds without taking any measures to recover the process delay. In this case, it is necessary to provide a prospect that indicates whether or not the project can be completed within the original delivery deadline (i.e., it is possible to meet the deadline) 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. However, EVA cannot show any countermeasures which it is possible to restore the project schedule to its original schedule and to meet the deadline even if such countermeasures exist. Therefore, the authors propose a method to estimate SEAC (Schedule Estimate At Completion) and EAC (Estimate At Completion) by taking account of the constraints on assigning human resources and show the estimation results of SEAC and EAC. The authors also propose a method that can automatically generate an alternative development plan which can recover the process delay if there is a countermeasure at least when it becomes clear that it is not possible to meet the deadline if the project proceeds without taking any measures to recover the process delay. With this method, it is possible to show the prospect that indicates whether or not it is possible to meet the deadline, and is also possible to propose a countermeasure or an alternative development plan that make it possible to meet the deadline.

This paper consists of the following sections. Chapter 2 describes the concept of EVA and introduces a concrete example of project (referred to as a sample project hereafter) that uses EVA to demonstrate the future cost and progress. This chapter also discusses three functions (F1), (F2), and (F3) for evaluating the project progress status and indicates that EVA can implement (F1) but cannot implement (F2) or (F3). In Chapter 3, the authors propose the constraints-based method as a means to implement the three functions mentioned in Chapter 2 and specify the implementation method. In Chapter 4, we apply our constraints-based method to the sample project and analyze the project to estimate the SEAC (Schedule Estimate At Completion) and EAC (Estimate At Completion). The estimation results show that the function (F1) required to evaluate the project progress status mentioned in Chapter 2 is achieved. Then, we compare the estimation results with those of EVA to clarify that the constraints-based method can generate more accurate estimation for the SEAC (Schedule Estimate At Completion) and EAC (Estimate At Completion) of the project. In Chapter 5, we present the project plan created by using the constraints-based method and show that the plan can be applied to the project to recover the process delay and meet the deadline. The results show that the functions (F2) and (F3) required to evaluate the project progress status mentioned in Chapter 2 are achieved. As a result, we conclude that our constraints-based method can be used to evaluate the project progress status. Chapter 6 compares this study with relevant studies. Section 7 describes the conclusion.

II. The Concept of Earned Value (EVA)

A. 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 (Planed orking 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.

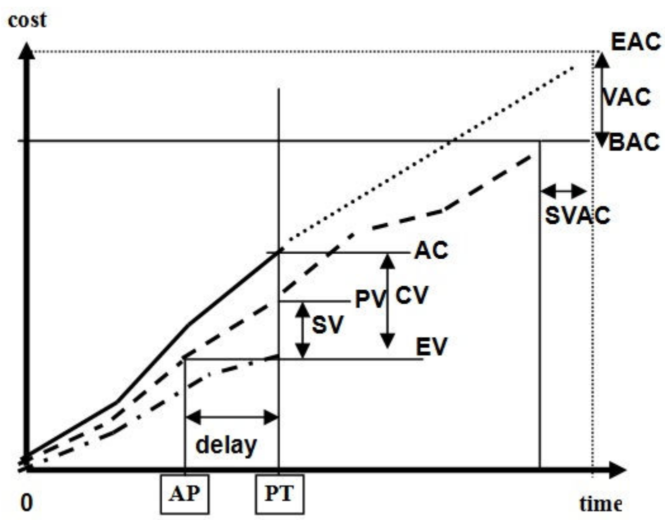

Fig. 1 A conceptual diagram of EVA

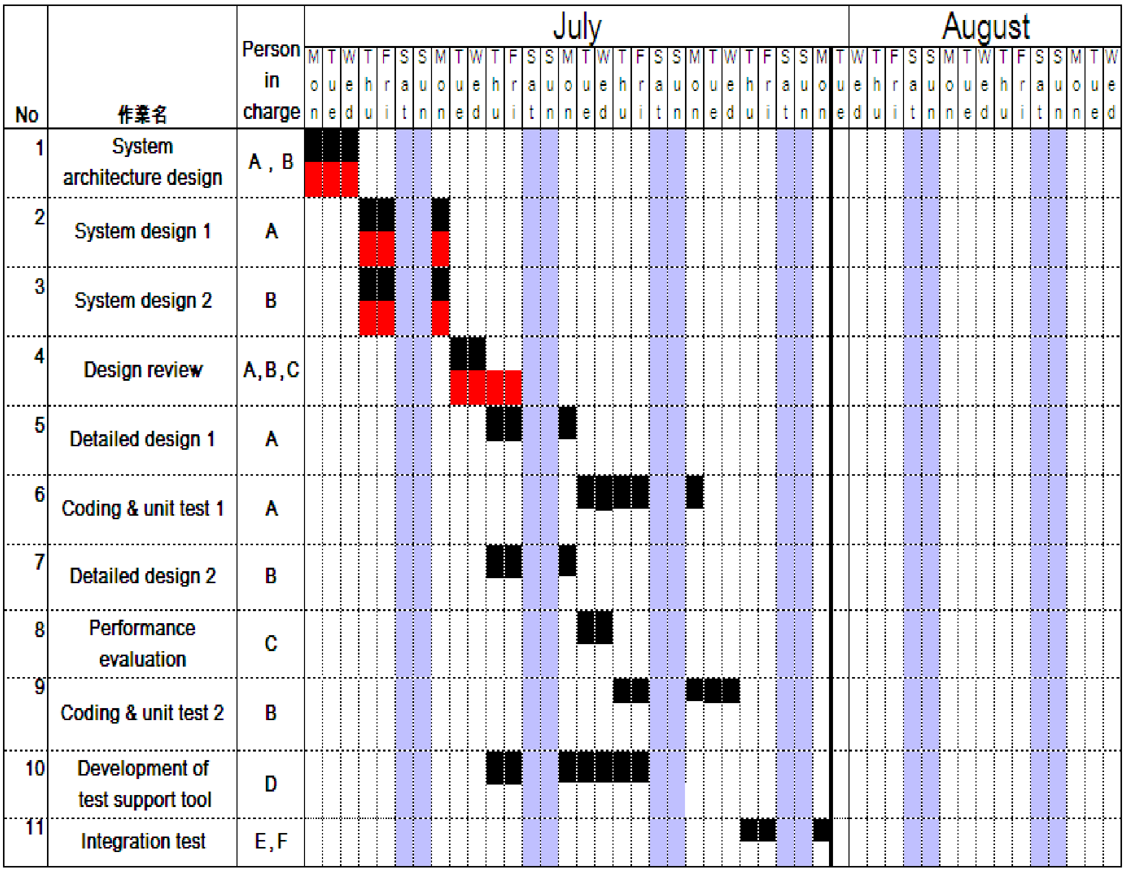

Figure 2. The original schedule and the status in the case of process delay

B. 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 Analysis (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 , and 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 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.

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.

Table 1. Human resource information

Worker	Skill	Rank of skill	Available period	Cost per hour (Yen)
A	System Analysis	a	-7/25	2500
	Software Design	b		
	Programming	c		
B	System Analysis	b	-7/25	3500
	Software Design	a		
	Programming	a		
C	System Analysis	a	-7/18, 8/1-	2800
	Software Design	b		
	Programming	c		
D	Software Design	b	7/10-7/25	3000
	Programming	a		
E	Quality Assurance	b	7/25-	3200
F	Quality Assurance	a	7/25-	3600
G	System Analysis	b	7/10-7/20	3600
	Software Design	a		
	Programming	a		
H	System Analysis	a	7/13-7/30	3600
	Software Design	a		
	Programming	b		

C. *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).

The following shows how to evaluate the progress status of the sample project with EVA.

(1) Calculation of PV

The following illustrates how to calculate PV for the SA process. The same calculation procedure can be applied to other processes. Since two Workers A and B are assigned to the SA process as shown in Figure 2 and the unit prices of A and B for the process is 2,500 yen and 3,500 yen, PV is calculated as follows:

The PV value for the SA process = (2500 + 3500) × 8 × 3 = 144,000

The PV values for the SD1, SD2, and DR processes calculated in the same way and the total values are listed in Table 2.

(2) Calculation of BAC

The PV value can be calculated from Figure 2 and Table 1 for each process. Since BAC is the total value of PVs for all processes:

BAC=144,000+60,000+84,000+140,800+60,00+100,000+84,000+44,800+140,000+168,000+163,200＝1,188,800

(3) Calculation of EV

The EV value is calculated for each process as the product of "the PV value of the process" and "the progress rate of the process."

Table 2 shows the EV value for each process and the total value when the DR process is completed, in which the process delay has occurred.

(4) Calculation of CPI (Cost Performance Index) when the DR process has completed

CPI is calculated by the following equation using the total values of EVs and the total values of ACs when the DR process has completed:

CPI＝EV÷AC＝358,400÷582,600≒0.615

(5) Calculation of EAC when the DR process has completed

EAC is calculated by the following equation using BAC, CPI when the DR process has completed, and the total values of ACs and the total values of EVs when the DR process has completed:

EAC ＝ AC+{ (BAC − EV) ÷ CPI }
≒582600+ (1,188,800 – 358,400)÷0.615
≒582600+830,400÷0.615
≒582600+1,350,244 ＝ 1,932,844

(6) Calculation of VAC (Variance At Completion)

Since CV (Cost Variance) is defined as "CV＝EV− AC," VAC (Variance At Completion) is calculated by the following expression using EAC and BAC where EAC represents the total value of EV BAC represents the total value of PV until completion of the project.

VAC ＝ EAC− BAC ＝ 1,932,844− 1,188,800 = 744,044

(7) Calculation of SPI(Schedule Performance Index) when the DR process has completed

SPI is calculated by the following equation using the total values of EVs and the total values of PVs until the DR process has completed:

SPI ＝ EV÷PV ＝ 358,400÷428,800 ≒ 0.836

(8) Estimation of SAC (Schedule At Completion) calculated when the DR process has finished

At the end of the DR process, SAC is calculated by the following expression using the initially planned working days and the actual SPI.

SAC ＝ (Initially planned working days) ÷ **SPI ＝ 21÷0.836 ≒ 25**

(9) Calculation of SVAC (Schedule Variance At Completion) at the end of DR process

The SVAC at the end of DR process is calculated as the difference between the SAC estimated at the end of DR process and the initially planned actual working days.

SVAC at the end of DR process = 25− 21 ＝ 4 (a delay of four actual working days)

The result of analysis (estimation) using EVA shows that the VAC (Variance At Completion) is **744,044** yen and the process delay is four working days, i.e. the SAC (Schedule At Completion) is August 3. Thus, 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.

(F1): The function to estimate SEAC (Schedule Estimate At Completion) and EAC (Estimate At Completion) in case that the project proceeds without taking any measures to recover the process delay when a process delay occurs.

(F2): Suppose that it turns out to be impossible to meet the original delivery deadline by applying the function (F1). In this case, the function (F2) determines whether or not the project can be completed within the original delivery deadline (i.e., it is possible to meet the original delivery deadline) if the project proceeds without taking any measures 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 turned out that it is not possible to meet the deadline.

As discussed above, EVA can provide the function (F1). However, EVA cannot provide a prospect that indicates whether t or not it is possible to meet the original delivery deadline, or cannot present a concrete development plan to restore the project to its original timeline even if such countermeasures exist. 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.

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

A. 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. For example, Process b in Figure 3 can be explained as follows.

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." For example, programming language, system testing, debugging and other processes require the personnel who have respective competencies. As a result, the software development work schedule depends on the constraints on human and non-human resource assignment conditions of relevant tasks for the software development.

(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. By implementing these constraints, the working rate can be used as a scale for evaluating a worker's workload and for checking if the worker is overloaded. The same concept is also applied to non-human resources. In general, capacity (upper limit of working rate) is likely to vary pursuant to the rank of such resource. Although the upper limit of assignable working hours per day is set to eight (8) hours, the default working rate in this study is set to 80% with consideration to work breaks and other intermissions.

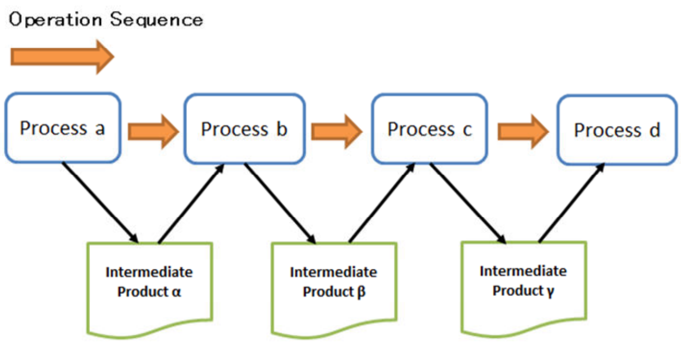

Figure 3. Constraints on the operation sequence of tasks based on the intermediate products

B. The mechanism for automatically generating the development plan based on the constraints

In References [3-11], creating a software development plan is regarded as solving a Combinational Optimization problem with many constraints. Based on this concept, we developed a system that can automatically generate a software development plan by employing the genetic algorithm, proposed the methods to construct the system, and proved the effectiveness of the system.

Figure 4 shows the mechanism to automatically generate a software development plan. As shown in Figure 4, the two-tiered model is employed in the GA (Genetic Algorithm) programming. The arrangement of genes in the upper layer represents the execution sequence of software development tasks (i.e. Constraint (C1)), and the arrangement of genes in the lower layer represents the assignment pattern of personnel to each process. (While only one person can be assigned to one process in References [3-6], the two-tiered GA model shown in Figure 4 is improved to allow more than one person can be assigned to one process in References [7-11].) The correspondence between the process and the assignment pattern is represented by the relationship between the gene locus of the upper and lower layers (the position each gene is located). For example, Figure 4 represents an example in which Pattern 2 is employed for the personnel assignment pattern of the DT process. Then, the assignment pattern table shows that two days are required to complete this task when it is processed by two workers, one of which is in the A (Excellent) level and the other is in the C (Fair) level. The personnel who are eligible for assignment to the DT process are those who have skills and qualifications required to perform the DT process (in other words who satisfy the constraint (C2)) and the candidates are such personnel in the A level and the C level. The personnel who are not booked yet in the period of the DT process (i.e. who satisfy the constraint (C3)) can be selected. If there is more than one eligible person, the most appropriate person is selected according to the selection policy employed by the system user, for example, the person whose unit price is the lowest. In addition, if more than one task is assigned to a single person in one day, only the plans that satisfy the constraint (C4) are eligible. Our system uses the genetic algorithm to generate candidate solutions one after another and selects the ones that satisfy constraints (C1) – (C4) as the solution. In other word, our system has employed the “Generate & Test” mechanism.

C. 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 account of 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.

D. 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.

C)Recovery of progress by holiday works

This is a method to shorten a process period by allowing holiday work for particular processes.

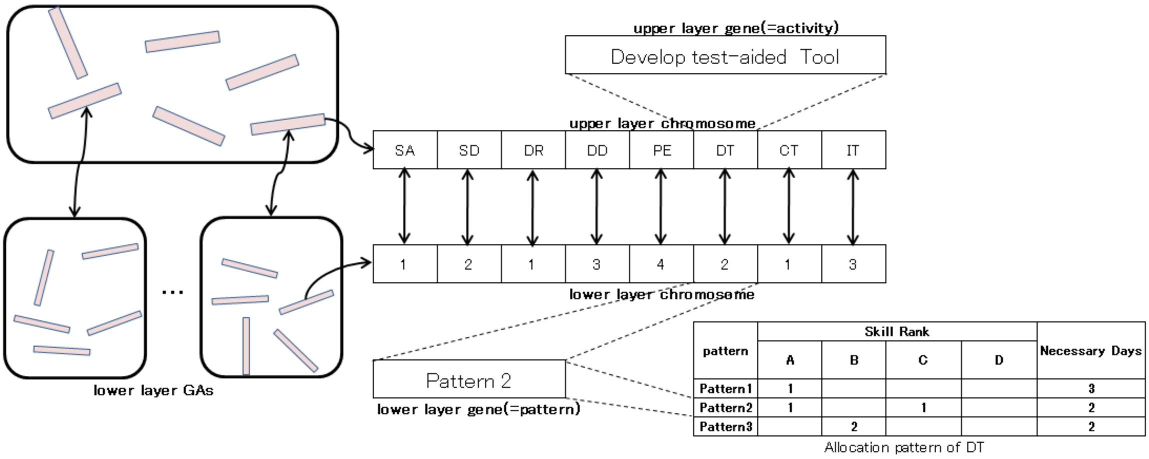

Figure 4. The structure of two-tiered GA

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.

Recovery of progress by holiday work employs the constraints (C2) – (C4) as they are and adds the constraint "(C5) Constraints on worker assignable days" in which holidays are regarded as the days not available for worker assignment, and allows holiday work by loosely applying the constrain (C5) to particular processes.

IV. ESTIMATION OF SEAC (SCHEDULE ESTIMATE AT COMPLETION) AND EAC (ESTIMATE AT COMPLETION)

A. 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).

This result shows that the estimated value of SEAC may vary depending on if it is calculated using the actual working days by EVA or if it is calculated using the constraints-based method.

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. As a result, the SEAC estimated by EVA is August 3 and the SEAC estimated by the constraints-based method is August 14. 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.

The systems we have developed so far did not provide the function to estimate EAC (Estimate At Completion), which is one of the important functions for evaluating the project progress status[3-12]. Now we added a new function to use a table such as Table 3 to estimate EAC. The EAC calculated by this function is 1,329,600 yen. On the other hand, the EAC calculated by EVA is 1,932,844 yen as described in Chapter 2. Although the SEAC estimated by the constraints-based method is six days longer than the SEAC estimated by EVA, the EAC estimated by constraints-based method is lower than the EAC estimated by EVA.

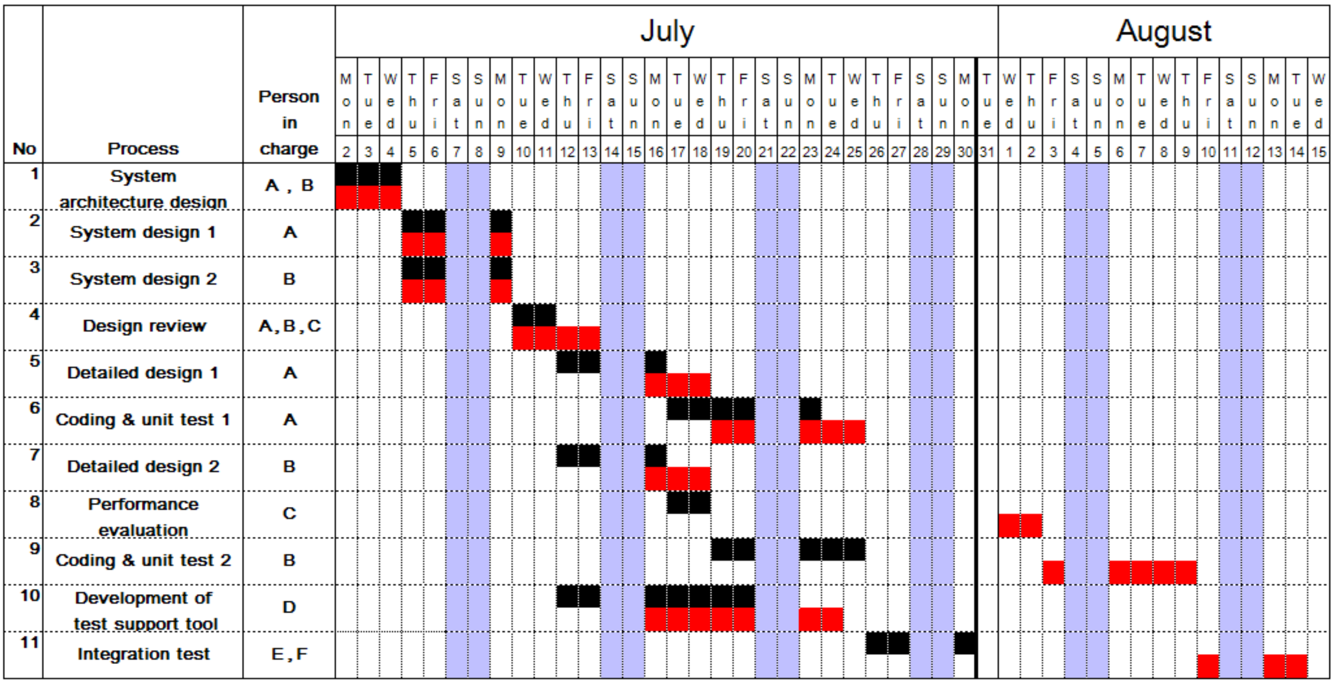

Figure 5. The analysis result of repercussions from process delays

It may seem strange, however the EVA is calculated under the conditions of assignable period of resources in which performance estimation (PE) by Worker C, coding and unit test 2 (C/UT2) by Worker B, integration test (IT) by Workers E and F are not performed before intermediate delivery and all of these tasks are performed after intermediate delivery of the preceding process.

Table 3. The table used to calculate EAC by taking account of the constraints when a process delay occurred

	A	B	C	D	E	F	Total
SA	60,000	84,000					144,000
SD1	60,000						60,000
SD2		84,000					84,000
DR	80,000	112,000	89,600				281,600
DD1	60,000						60,000
DD2		84,000					84,000
DT				168,000			168,000
PE			44,800				44,800
C/UT1	100,000						100,000
C/UT2		140,000					140,000
IT					76,800	86,400	163,200
Total	360,000	504,000	134,400	168,000	76,800	86,400	1,329,600

V. Automatic Generation of Development Plan by Constraint-based Method for Recovering a Process Delay

As the result of the perturbation-based repercussion analysis described in Chapter4, 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. In this paper, we introduce the method that can generate a countermeasure by Holiday work as we have completed implementation of the system. We now provide the systems for all three functions described above. With these systems, it is possible to use one of these methods of Crashing, Fast-tracking, or Holiday work or their combination to generate a countermeasure plan to recover a process delay. This indicates that all of the means to recover a process delay are examined and it is possible to check to see if there is any countermeasure plan to recover the process delay. It also indicates that it is possible to present a concrete countermeasure plan to recover the process delay, if such a plan exists. Now, the functions (F2) and (F3) are available for evaluating the project progress status.

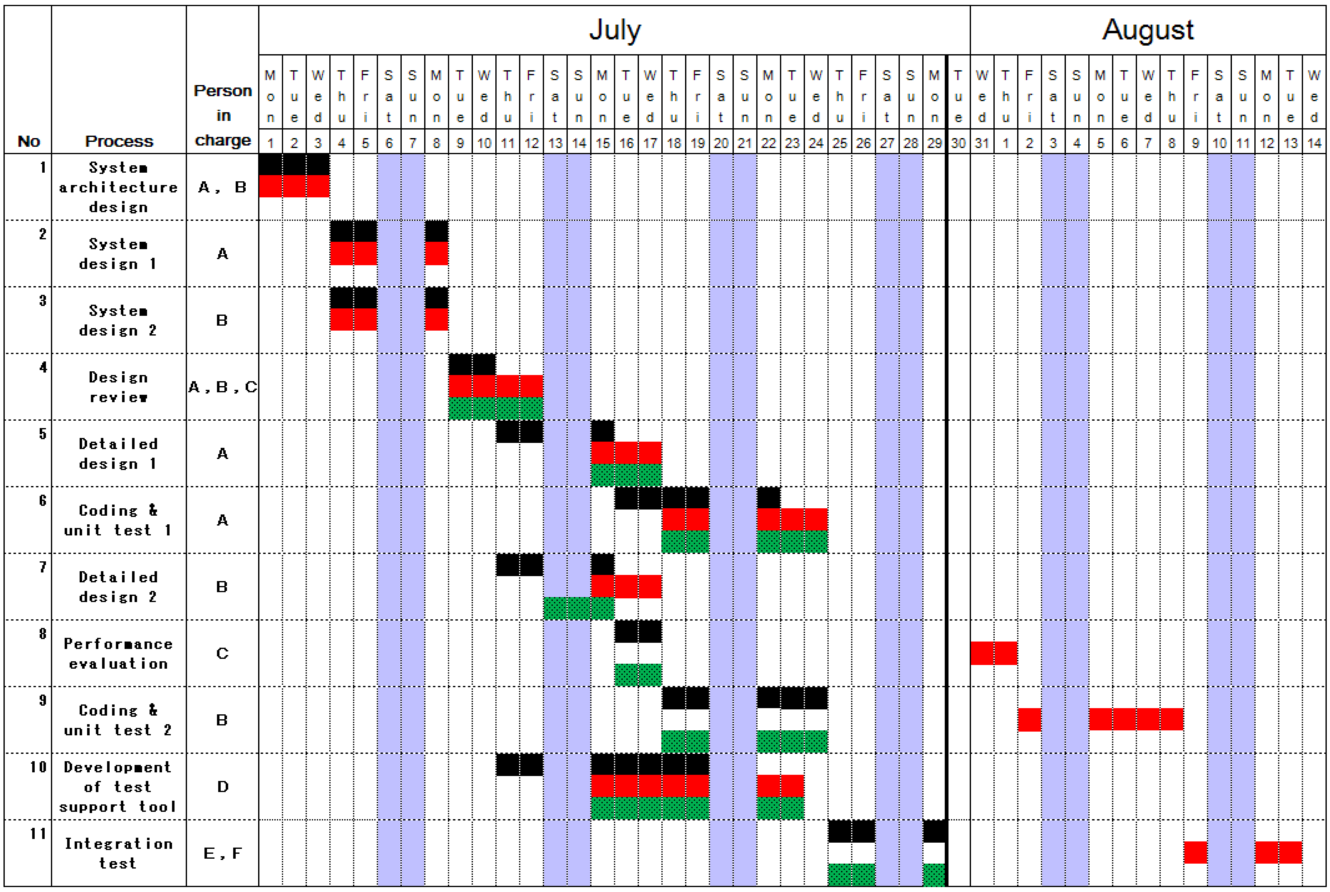

Figure 6. A countermeasure schedule plan by holiday shift

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).

A. 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. This indicates that applying Crashing alone cannot complete the project by the initially set deadline. 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.

B. 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. This indicates that applying Fast-tracking alone cannot complete the project by the initially set deadline. 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.

C. Countermeasure plan by Holiday work

We tried automatically generating a countermeasure plan by Holiday work to recover the process delay. As a result, more than one countermeasure plan has been generated. Among these plans, only one plan could be used to complete the project by the delivery deadline initially set.

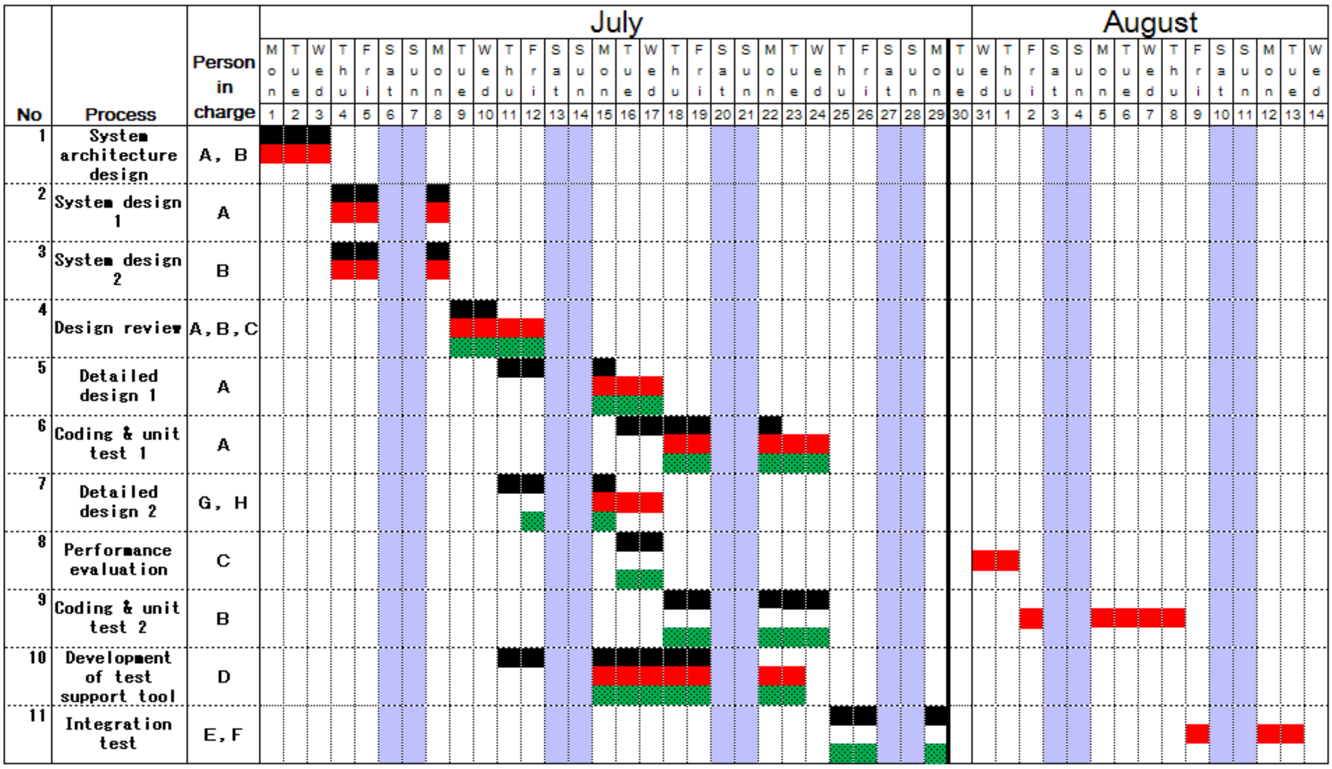

Figure 7. A countermeasure plan for process delay generated by combination of Crashing and Fast-tracking

This plan is listed in the bottom row in Figure 6. This plan indicates that Worker B can complete the DD2 process by working for three days from July 14 to 16 including holiday work. (Worker B has two compensation days on July 17 and 18.) As a result, Worker C can start the PE process on July 17 as initially scheduled. This result indicates that all of the processes can be completed by July 30 if Worker B works on holidays on July 14 and 15 and has two compensation days on July 17 and 18. 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 1,329,600 yen.

VI. 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, 21, 22] getting attention recently in comparison with our approach. We have to explain TOC (Theory of Constraints) [19, 20] before discussing CCPM. TOC is a management method that focuses on the weakest portion (Constraint Conditions in TOC terms) in company activities and reinforces and improves that portion intensively to maximize results with minimal effort. Based on the idea of TOC, CCPM, as a project management method, performs optimization from the viewpoint of entire project. CCPM uses Critical Chain in place of conventional Critical Path and removes the additional part of efforts included in the estimation phase for safety purpose to shorten each process period (specifically, adopts an effort estimation of 50% success probability), and then add a project buffer (margin days) at the end of a process on a Critical Path to manage the entire process at a single point. In addition, it inserts a joining buffer between the tasks on the Critical Chain path and the tasks on the path that joins to the Critical Chain to prevent the Critical Chain from being affected by delays of tasks on the path that joins to the Critical Chain path. Then the project schedule is created by taking account of the delivery date, cost and constraints of the resources. The project manager then understands the progress of the whole project by examining the consumption ratio of the buffer instead of managing the progress of each process. The above is an outline of CCPM.

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.

VII. 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) estimated by EVA for the sample project. Then we pointed out that the following three functions are required to evaluate the project progress status.

(F1): The function to estimate SEAC (Schedule Estimate At Completion) and EAC (Estimate At Completion) in case that the project proceeds without taking any measures to recover the process delay when a process delay occurs.

(F2): Suppose that it turns out to be impossible to meet the original delivery deadline by applying the function (F1). In this case, the function (F2) determines whether or not the project can be completed within the original delivery deadline (i.e., it is not possible to meet the original delivery deadline) if the project proceeds without taking any measures 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 turned out that it is not possible to meet the original deadline.

Then, we pointed out that that EVA can implement only (F1) but cannot implement (F2) or (F3) and that EVA alone is not enough to evaluate the project progress status.

We proposed the constraints-based method to provide the above three functions and clarified how to implement the method. Then, we applied our constraints-based method to the sample project and analyzed the project to estimate the SEAC (Schedule Estimate At Completion) and EAC (Estimate At Completion). We used the estimation results to show that the function (F1) required to evaluate the project progress status is achieved. Then, we compared the estimation results with those of EVA to clarify that the constraints-based method can generate more accurate estimation for the SEAC (Schedule Estimate At Completion) and EAC (Estimate At Completion) of the project.

We discussed that we can apply three two methods, (a) Crashing, (b) Fast-tracking, and (c) Holiday work, and their combination as the means to change the development plan (development schedule and personnel assignment) to recover the process delay. Then, we showed that it is possible to show whether or not any countermeasure plan exists by employing the constraints-based method and using any one of the three methods or their combination, and automatically to generate concrete countermeasure plans in which the development schedule and personnel assignment have been modified, if such a plan exists at least. 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 concrete countermeasure plans that complete the project by the initially set delivery deadline by employing the Holiday work method or 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 showed the effectiveness of the method.

References

[1] Project Management Institute, Inc., "A Guide to the Project Management Body of Knowledge:(PMBOK Guide) 4th Edition," 2008.

[2] Quentin W. Fleming & Joel M. Koppelman, Earned Value Project Management, Second Edition, 2002.

[3] Seiichi Komiya, Naota Sawabe , and Atsuo Hazeyama, "Constraint-Based Schedule Planning for Software Development," IEICE Trans. Inf. Syst., Vol.J79-D-1, No.9, pp.544-557, 1996.

[4] S. Komiya, and A. Hazeyama, " A Meta-Model Framework for Software Project Management System," IEICE Trans. Inf. Syst., Vol.E81-D, No.12, pp.1415-1428, Aug. 1998.

[5] A. Hazeyama, S. Komiya, " Workload Management Facilities for Software Project Management." IEICE Transaction Information Systems, Vol.E81-D, No12, pp.1404-1414, Dec 1998.

[6] S. Komiya, and A. Hazeyama. " Projecting Risks in a Software Project through Kepner-Tregoe Program and Schedule Re-planning for Avoiding the Risks," IEICE Transaction Information Systems, Vol.E83-D, No4, pp6275-639, April. 2000.

[7] D. Kinoshita, R. Yaegashi, H. Hashiura, K. Uenosono, and S. Komiya, "An Automatic Schedule Planning System," Strategies and Evaluation for Implementing the System, Joint Conference on Knowledge-Based Software Engineering 2004(JCKBSE'04), Protvino, Russia, pp.37-48, Aug. 2004.

[8] Rihito Yegashi, Daisuke Kinoshita, Hiroaki Hashiura, Kazuhiro Uenosono, Yuichirou Hayashi, and Seiichi Komiya, "Automatically Creating a Fast-Tracking-Based Countermeasure Plan against Process Delay," IEICE Trans. Inf. Syst., Vol.J88-D-1, No.2, pp.215-227, 2005.

[9] Kiminori Takasuka, Shougo Shimamura, Hiroki Uchikawa, Daisuke Kinoshita, Yuichirou Hayashi, and Seiichi Komiya, "Automatically Creating Software Development Plans to Have Taken Slack Time into Consideration," IPSJ, vol.48, No.8, pp.2713-2724, Aug. 2007.

[10] Yuichirou Hayashi, Daisuke Kinoshita, Hiroaki Hashiura, Kiminori Takasuka, Rihito Yaegashi, and Seiichi Komiya, "Facilities for Automatically Creating the Profiles of Guest Workers Required for Creating Software Development Schedule Plans," IEICE Trans. Inf. Syst.,

[11] D. Kinoshita, R. Yaegashi, K. Uenosono, H. Hashiura, H. Uchikawa, and S. Komiya: "Automatic Creation of a Crashing-Based Schedule Plan as Countermeasures against Process Delay," International Jouranal of Systems Applications, Engineering and Development, Issue 4, Vo. 2, pp. 170-177, 2008.

[12] Daisuke Kinoshita, Hiroki Uchikawa, Yuya Kosaka, Seiichi Komiya: "Automatic Detection of Bottleneck Processes through Perturbation-based Repercussion Analysis," IPSJ, Vol.50, No.12, pp. 3084-3094, Dec. 2009."

[13] Leach, L.P., “Critical Chain Project Management,” Artech House Professional Development Library, London 313, 2000.

[14] Penedo, M.H. and Stuckle, E.D., “PMDB – A project master database for software engineering environments,” 8th International Conference on Software Engineering, pp.150-157, 1985.

[15] Liu, L. and Horowitz, E., “A formal model for software project management,” IEEE Trans, Softw. Eng., Vol.15, No.10, pp.1280-1293, 1989.

[16] Liu, L. and Horowitz, E., “Object database support for a software project management environment,” ACM SIGSOFT Software Engineering Notes, Vol.13, No.5, pp.85-96, 1988.

[17] Y. Matsumoto, and T. Ajisaka, “A data model in the software project database KyotoDB, ” JSSST Advance in Software Science and Technology 2, pp.103-121, 1990.

[18] H. Sato, “Project management expert system,” Proc. ACM CSC’87, Feb. 1987.

[19] Kimio Inagaki, "TOC: a Revolution of Project Management by using Critical Chain," Management Center, Japan Management Association, 1988(in Japanese).

[20] Eliyahu M. Goldratt, "THE GOAL", North River Press, 1984.

[21] Eliyahu M. Goldratt, "Critical Chain", North River Press, 1997.

[22] Hidetaka Nakajima, and Koji Tsumagari, " Project Management by using Critical Chain," Management Center, Japan anagement Association, 2003(in Japanese).