Optimization of Grouping and Team Formation of Students for Class Exercise Based on Analysis of Human Factors by Covariance Structure Analysis

KIYOMI SHIRAKAWA, HIROAKI HASHIURA, HITOMI SAITO, SEIICHI KOMIYA
Graduate School of Engineering, Shibaura Institute of Technology
Komiya Lab, 13F, Kenkyu-to, 3-7-5 Toyosu, Koto-ku, Tokyo
JAPAN

{m706105, m705107, m106044, skomiya}@shibaura-it.ac.jp
http://www.komiya.ise.shibaura-it.ac.jp/

Abstract: - Group exercises for software development adopting more practical approaches are conducted in Department of Information Science and Engineering, Shibaura Institute of Technology in order to allow students to acquire knowledge and skills for software development. (Such student groups can be recognized as "teams" in a precise sense because each student member belonging to the group takes certain partial charge of software development.) But the class time for students acquiring software engineering is limited. That is why the support system, which is called EtUDE (Environment for Ultimate software Development Exercise), has been developed and introduced to help students to take practice together at any time and any place. However, differences in students' abilities for software development have sometimes led to problems like that some teams can not achieve a given task by due date. To solve the problems, Hashiura, et al. [1] developed the system, EtUDE/GO which automatically decides an optimal scheme for grouping and team formation for class work. In spite of this, human factors influencing on the team formation remained unknown. Therefore, the authors introduced a covariance structure analysis and so on to find a relational expression between true factors influencing directly on team formation and alternative characteristics having effect indirectly on it. Applying the obtained relational expressions to EtUDE/GO, they produced an optimal team formation scheme automatically. Then, how the optimization of team formation exerted influences on the class exercises by measuring each student's contribution evaluated with log information of their work. The result of the analysis showed that the students who were expected to make contributions for achievement of tasks and who were disposed in each team assisted their team well so as to complete the tasks in actual class exercise as they were expected. As a result, all the team was confirmed to have achieved the assignment without delay. In fact, this paper shows it was confirmed that the optimization of team formation enabled all the teams to complete the exercise assignment by due date.

Key-Words: - Optimizing Project Team Formation, Exercises in Units of Groups, Exercise for Software Development, Genetic Algorithm, Covariance Structure Analysis, Factor Analysis, Path Diagram, Maximum Likelihood Estimation, Chi-Square test

1 Introduction

Department of Information Science and Engineering, Shibaura Institute of Technology conducts a 6-month exercise course of "software development" for senior students. The purpose thereof is to allow the students to learn knowledge and skills necessary for development work with experience of all the process of software development from requirement extraction to programming as making project teams comprising several students. (In the course the small organization including several people is regarded as a project team, so the course provides not group exercise but project exercise.) The course is carried out with the supporting environment for software development course, EtUDE (Environment for Ultimate software Development Exercise) [2], which was developed in order to improve efficiency of students' work to include functions such as communication support, deliverable management support, project management support, and a function to collect automatically all the information which students generate.

This project exercise adopts the exercise form where each member belonging to a team takes partial charge of a project and cooperate each other to solve the task; therefore, project size is rather large and work flexibility of students is high.

Naturally, if ability differences among the teams are not retained small, qualities of their deliverables would vary greatly. In addition, if there is not at least one member per each part of the project who is able to share the part, the team would possibly not complete the task within a time frame. Hashiura, et al. [1] saw the above problem and defined the condition (limiting condition) to optimize the team formation as follows:

(C1) Each roll of each team should have at least one student who can perform the roll.

(C2) Ability differences among teams should be as small as possible.

(C3) Head-count difference should be within one person.

The following condition was also added from educational consideration to provide student who can not perform any roll with opportunity to learn.

(C4) Even students who do not have aptitude to take any roll should be disposed in any one of the teams.

Next step is to research human factors which each student has in order to determine whether they have the aptitude to perform the given work or not. Hashiura, et al. [1] considered that there were 2 kinds of human factors influencing on the team formation.

(F1) True factors (also called "true characteristics")

We call the factors (also called "characteristics") as "true factors", which seem to influence directly on students' ability to perform each roll necessary for software development

(F2) Alternative characteristics

Among the factors which seem to influence indirectly on students ability to perform each roll necessary for software development, we especially call what can be measured directly as alternative characteristics.

The true factors can rarely be measured directly, so that it is necessary to take a way to control the true factors by picking out values of the alternative characteristics and controlling them. For that, relations between the true factors and the alternative characteristics must be formulated into a relational expression. Hashiura, et al. [1] developed the system, EtUDE/GO. If objective variables (the true factors) can be represented with an expression of explanatory variables (alternative characteristics), the system generates an optimal team formation scheme with the expression.

However, the research of Hashiura, et al. [1] did not disclose relations between the true factors and the alternative characteristics.

This paper represents the objective variables with an expression of explanatory variables by disclosing the relations between them using the CSA (covariance structure analysis), a method of multivariate analysis. It then, generates an optimal team formation scheme by applying the expression to EtUDE/GO. Actual team making is conducted based on the generated optimal team formation scheme to carry out the exercise class. In the end of the 6-month course, influences (effects) brought by the optimization of team formation are analyzed and evaluated.

The structure of this paper is detailed as follows. Chapter 2 explains the functional overview of EtUDE which was developed in order to solve the problem of insufficient class hour. Necessity of developing EtUDE/GO is also described from result of student evaluation. Chapter 3 shows relational researches for grouping. Chapter 4 shows how to implement an optimal grouping and team formation in a class exercise. Chapter 5 shows how to achieve EtUDE/GO with introduction of genetic algorithm. Formulation of a hypothetic model and data collection are described in Chapter 6 and the model's verification and minor amendments are instructed in Chapter 7. Chapter 8 depicts evaluation of the team formation and its relevance. Chapter 9 demonstrates the effectiveness evaluation of the optimization of team formation after the course. Chapter 10 concludes this paper.

2 Functional Overview of EtUDE

Hashiura, et al. [1] considered that a support system with which each student can use without time or place limitation was needed in order to solve the problem of insufficient class hour, and constructed EtUDE as a Web application system. It includes "communication support function" as a support function for members belonging to the same group and "deliverable management support function" and "project management support function" as support functions for allowing plural members to cooperate to generate one deliverable even if they work separately.

(1) Communication support function

This is the support function enabling members of the same group to work together even if they are in different places and to use the system different time.

(i) Questions and answers for them.

(ii) Suggestions and opinion exchange thereto.

(iii) Communication and confirmation from professors.

For supporting the above three communication formats, there are two types of message boards: one is exclusive message board for each group and the other is Q&A and communication message board for the use of all the students. Both provide tree-view and thread-type message boards. The tree-view message board has the advantage that comments can be exchanged by related topics (threads). Additionally, it has the visual benefit to understand which comment a reply related with easily due to its tree view.

In the exclusive message boards for groups, students can refer uploaded deliverables with the following "deliverable management function", which is a different point from the Q&A message board.

Those message boards include the function which notifies object persons that a message is left at the same time when it is written in the board. This function aims at awareness.

(2) Deliverable management function

In software development, there are many kinds of works such as development planning, analyses, design, coding, and test, which leads to generation of deliverables such as planning document, requirement specifications, design specification, test cases, source codes and minutes.

Among these deliverables, there are the ones which become factors to decide relations between preceding and succeeding work (i.e. work order thereof) like the case where a subsequent operation is conducted as referring to a (intermediate) deliverable made in its precedent operation. Such (intermediate) deliverables should be shared not only within the group making or using them but also with professors who figure out progress of the exercise and give the students proper advice. In addition, when there are many groups, it is quite a burden for the professors to receive and manage all the deliverables from the students for progress comprehension and scoring. Therefore, it is necessary to adopt a system consolidating all the deliverables electronically and allowing the professors as well as the students to access thereto at any time and any place. On the ground of the above, this includes "deliverable management function", "module management function" and "source code management function".

(3) Project management support function

The project management support function includes "report writing support function" and "failure management function".

(i) Report writing support function

This function provides templates of items which the professors want to know for the students who do not see what they should report. The students can report the information which the professors want to hear by collecting information following the provided templates. Moreover, the information can be unified for management by the function. Development plans, work item management, work reports, minutes, and development completion reports are intended here. Among the above, work item management is the report to comprehend progress of work items and important for project management.

(ii) Failure management function

This function (bug tracking function) manages bugs generated in coding at phases of implementation and test. Adequate tests after coding almost always discover bugs. Basic strategy to handle the problem is to list the bugs found by the time to debug them in order of priority. The list of bug makes it possible to clarify roll sharing of debug work and to manage progress degree thereof.

(4) Software development support

This provides the functions which individuals need to develop software. The course requires the students to develop a program for a web application system for "meeting room reservation system" as their task with Java by analyzing and designing with OOSE (Object Oriented Software Engineering). Therefore, it is possible for the students to use JUDE (system design support tool) to support for analysis and design processes and Eclips (program developing environment) to support programming process, for example.

3 Relative Researches

There are researches utilizing information of students' individual skills because consideration about such information is necessary to make groups for software development exercises. Hazeyama [4] and Hazeyama, et al. [5] conducted experiments by collecting attribute information for skill: system analyzing skill, interest level for system development, career options, leadership skill, and communication skill, to regard them as the attribute information about the skill which students have, so that groups can be made based on a strategy to minimize ability difference among groups as referring the information. The research pays attention to the information of student's individual skills but it does not go further to roll sharing based on the skill information. Plus, the grouping results could be influenced by the students' self-evaluation (subjective view) because the research uses evaluate values obtained by questionnaires using interval scale for collecting individual attribute information. [6]

This paper emphasizes that project team making needs consideration about role sharing based on students' individual skills and aptitudes, and evaluation thereof serving as a base of roll sharing needs the use of objective data.

4 Analysis Conducted in This Paper – Evaluation Process

4.1 Analytical Process of Human Factors

For the way to discover true factors, there are explanatory factor analysis to extract them explanatorily with factor analysis method etc. and confirmatory factor analysis to concrete relational expressions by improving a model in stages with given hypotheses. CSA [7] is developed version of the confirmatory factor analysis. It is not the best policy to use the confirmatory factor analysis because this method takes a lot of time to confirm whether all the major factors are enumerated in observable factors.

Therefore, the paper firstly uses not the explanatory factor analysis but the confirmatory factor analysis. The confirmatory factor analysis will be used for supplement of the explanatory factor analysis only when a stable hypothetic model can not be obtained.

CSA models a structure of problem and represents it with three variables: latent variable, observed variable, and error variable. This is also called a casual model because the model described by CSA represents causal relations among the variables.

The analytic processes are as follows:

(P1) To model a structure of problem with path diagrams.

(P2) To represent the relations among the three: latent variable, observed variable, and error variable, as a regression equation based on the path diagrams.

(P3) To improve the hypothec model as confirming how it is applicable by Chi-Square test and so on.

4.2 Automatic Generation of Optimal Team Formation Scheme and Confirmation of Limitation Satisfaction

A relational expression between observed variable and latent variable is set up and the latent variable is represented with an expression of the observed variable. Next, observed data of alternative characteristics are substituted into EtUDE/GO to generate an optimal team formation scheme with the use of the expression obtaining value of the observed variable from the observed data of the alternative characteristics. At this time, this generated optimal scheme is verified whether to satisfy the conditions (C1) to (C4) defined by Hashiura, et al. [1]. In addition, only the condition (C2) is done to confirm that there are no ability differences among the teams by using ANOVA. The other conditions are confirmed manually.

As soon as those conditions are confirmed, team making is conducted based on the scheme to allocate the students to each team for class exercise.

Table 1 shows the course overview of 2006.

Table 1 Course overview

2006	
Exercise Task	Development of meeting room reservation system
Exercise Term	November 16th, 2006 to January 9th, 2007
Number of Objective Students	57 people
Number of Team Members	4 to 5 students
Number of Teams	14 teams

4.3 Verification of Effects of Optimization of Team Formation

After the exercise term of software development, influences (effects) which the optimization of team formation brought to this class exercise. In concrete terms, it is firstly checked whether all the teams could complete the task without delay or not. If the entire team achieved the task without delay, and what was brought by the optimization of teaming (effect), is verified. Speaking more specifically, it is verified whether or not the students who were expected to contribute the task achievement by being dispersed into each team in the course exercise, actually showed dedication by the optimization of team formation in each team to which they were allocated. A discrimination analysis is used for this verification.

In addition, the professors including TAs (teaching assistants) of the course determine whether or not each student actually took a helpful role in the software development exercise with reviewing the log information of the all students' work which is obtained by EtUDE automatically.

5. Automatic Generation of an Optimal Team Formation Scheme by Introducing GA

5.1 Reason for Introduction of GA

Genetic algorithm (hereinafter called GA) is the way to generate nominations of solution one after the other and then to decide on which solution to take by using an evaluation fitness function (objective function) thereto. Therefore, if the number of combinations increases, computation time also increases exponentially. However, this team-making problem concludes the small number of solutions which satisfy all the conditions; namely, there are a few objects to examine. So, it is possible to acquire an approximate solution fast. Additionally, it is possible to acquire an approximate solution with knowledge of object domain, because GA is given in a form of conditions (limiting conditions) which the solution should satisfy. This is another advantage.

5.2 Coding

In the case of optimizing team formation with GA, it is necessary to decide coding (order of chromosome and their meanings) and a gene manipulation algorithm.

Fig.1 shows the genetic coding which the system used.

The chromosome represents the order of students and genes store IDs of students. The length of the chromosome is fixed because it is the number of the students taking the course. Each locus is allocated for number of team to which the students belong and their role therein, and the roles are determined depending on the location of genes (locus). Further, other areas store alternative characteristics corresponding to each student and they can be referred when individual or team ability is evaluated.

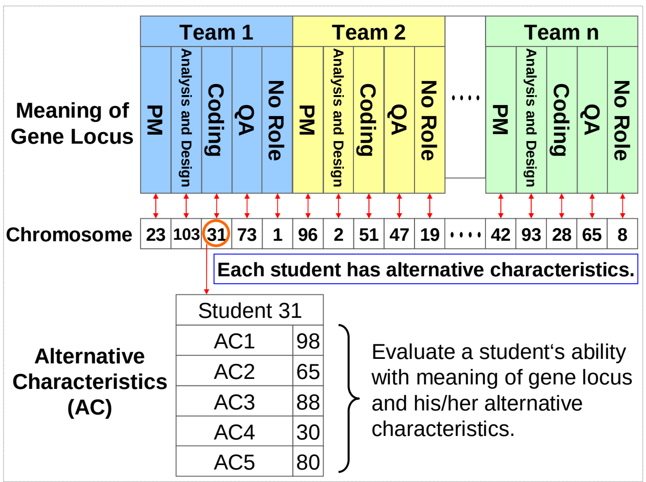

Fig. 1 Coding

5.3 Evaluation of Individual Ability

Hashiura, et al. [1] conducted an evaluation to determine whether or not a person has the ability to perform a role (aptitude) indirectly with the above-said alternative characteristics.

It is not an equal relation (1:1) between ability required to perform a role and alternative characteristics representing whether or not a student has the ability or not. Ability of a student carrying out a role i (student ID: l) can be represented as expression (1) in relation with a subject j (j=l to p) and grade thereto.

In this experiment, relation between object variables and alternative characteristics and significance degree of each role are obtained with covariance structure analysis to apply to the model.

$$y_{il} = \sum_{j=1}^{p} \alpha_{ij} x_{jl} + c_i \quad (1)$$

i : Role Number j : Subject Number
y_{il}: Objective Variable a_{ij}: Coefficients
x_{jl}: Students' Score C_i: Constants
l : Student Number p : Number of Subjects

5.4 Evaluation of Possibility of Work Performance

When allocation of roles to students is conducted, it is necessary to satisfy the limiting conditions showed in Introduction of Chapter 1: "each roll of each team should have at least one student who can perform the roll".

Determination whether or not a student can perform a role is done by reviewing if his/her ability necessary to carry out the role exceeds a certain level. Essentially, this evaluation of possibility of work performance can be judged by reviewing whether or not the expression (1) showed in 5.3 excesses a certain level.

5.5 Evaluation of team ability

No that Individual ability is calculated as described in 5.3; team ability is figured out with the value.

$$O_k = \frac{1}{p \cdot n_k} \sum_{i=1}^{n_k} w_i Y_{il_k} = \frac{1}{p \cdot n_k} \sum_{i=1}^{n_k} w_i \sum_{j=1}^{p} \left(\alpha_{ij} x_{jl_k} + C_i \right) \quad (2)$$

i : Role Number k : Team Number
O_k : Ability of Team k (Average)
l_k : Student Number in Team k
n_k : Number of Students in Team k
w_i : Importance of Role i in Team k
r : Number of Roles p : Number of Subjects

Examples of ability calculation adopted in our experiment are given in Table2.
About calculation method, as mentioned in 5.3, the individual ability value is represented by the sum of the alternative characteristics. About whether one is able to fulfill his/her role or not, as mentioned in this subsection, it is judged by whether the alternative characteristics, supposedly the most related with the role, is above a certain level or not (shadow-masked in Table 2).
Specifically, the ability value of the student with the register number of 001 and the role of the leader in Table 2 is equal to 276. Furthermore, since this student plays the role of a leader, the alternative characteristics supposedly the most related with the role, i.e. the score of problems on project management, must be above a certain value. And as his/her score of problems on project management is 84 while at present this level is defined to be available so long as the score is above 60, we can conclude that he is able to fulfill the role of a leader.

Table 2 Ability Calculation Examples

Role	Leader	Analysis and Design	Coding	QA	No Role
Student Number	001	002	003	004	005
The score of problems on project management	84	80	90	78	67
The score of problems on software design	69	90	75	80	76
The score of problems on programming	75	79	88	65	78
The score of problems on software testing	48	62	-	93	63
Individual Ability	276	311	253	316	284
Team Ability	1440 / 19 = 75.79				

However, if arranging students in this way, it might happen that some student of low ability could not be assigned to any team. Considering this, we prepared a role called “no role” so that even a student who is not able to fulfill any role could also be assigned to a team. To evaluate the ability of a member who is not assigned any role, consider the average of all alternative characteristics to be the supposed alternative characteristics. Besides, the supposed alternative characteristics are not used to judge whether one is able to fulfill his/her role or not.

5.6 Optimization of Team Formation

Optimization of team formation is conducted by making teams as satisfying the limiting conditions of possibility of work performance described in 5.4 and obtaining dispersion of ability as showed in 5.5 to minimize it.

In particular, a team formation scheme which meets the limiting conditions of possibility of work performance is made. Next step is to calculate the ability of all the team to compute dispersion of ability among the teams. This step is repeated several times until the student allocation to minimize ability dispersion among the terms can be done, which is the optimal team formation described by Hashiura, et al. [1].

The objective function O to obtain the optimal team formation is represented as follows:

$$O = \min\left\{ \frac{\sum_{k=1}^{m} \left(o_k - \overline{o}\right)^2}{m} \right\} \quad (3)$$

n : Number of All the Students
$\overline{o}$: Average Ability of All the Students
m : Number of Teams

5.7 Algorithms and Parameters Used in GA

This system adopts roulette wheel algorithm as selection algorithm, uniform crossover as crossover algorithm, and algorithm by replacement of 2 points as mutation. These algorithms do not generate lethal genes in the case of team making, so efficient team formation is possible.

Table 3 shows the parameters of the genetic algorithm used in this system.

Table 3 Parameters of genetic algorithm

Individual Size	2000
Crossing Probability	0.7
Mutation Probability	0.01
Generations to Calculation Convergence	4000

5.8 System Implementation

The core of our system, EtUDE/GO is implemented as a Web application scripted in Java language [8]. Being a Web application, it can be run independent of the platform on the client side, and it is also easy to accomplish a system upgrade. In

addition, EtUDE/GO is based upon Java Servlet 2.4 [9] specification, as well as JSP 2.0 [10] specification. Not limited to our operation environment this time, our system can be run on any Web application server given that it supports Java Platform Enterprise Edition [11] based upon the above specifications.
Besides, on implementation of our system, except the core system of EtUDE/GO, we have made positive use of open source software. Recently, open source software has been improved in both functionality and reliability, its technical gap with software on the market is being narrowed, and its introduction and operation are remarkably economic. Considering such merits as above, we adopted open source software.
In our experiment we used Winstone [12] as the Web application server, which is suitable for embedded applications and brings little overhead when being run. For the same reason, we chose MySQL[13] as the database.
As for the client side, since it can be accessed using a Web browser (Internet Explorer, Firefox, Safari etc.) contained in a standard OS, it is not necessary to install any special software.
On making use of this system, the user should enter plenty of score data into the system, in order to reduce the teacher's work we adopted the Microsoft Excel file as the interface file. In this way, the necessary data could be imported into the system collectively in one time.
As score data is handled in this system, full attention should be paid to its security. For this purpose, once score data is entered, it remains only inside memory and should never be saved in DB. Besides, teaming information exported from this system is restricted only to be identification IDs and team numbers of the students, while the students' score data can not be known from team formation information.

6 Formulation of a Hypothetic Model and Data Collection

As software tool for statistics analysis, AMOS (analysis of moment structure) is utilized. This can confirm and amend a hypothetic model visually with path diagrams.

6.1 Formulation Based on a Hypothetic Model

The hypothetic model of the human factors which influence on team formation for class exercise of software development is formulated. The model of Fig. 2 consists of several factors (true factors) which represent ability themselves to perform each role necessary for software development and several alternative characteristics.

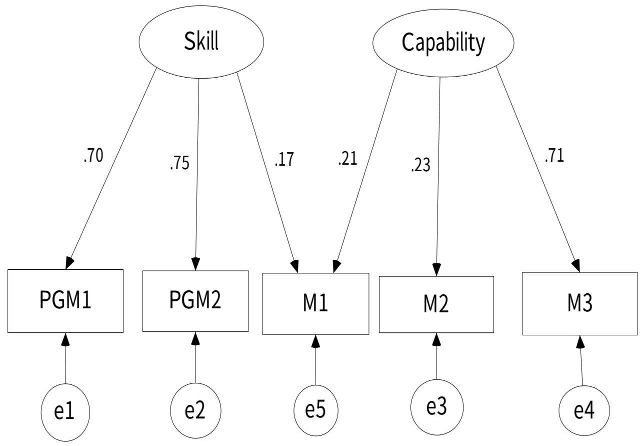

Fig. 2 Causal relation model between performance ability factors and alternative characteristics (standardized estimate values)

Larger scale of software to be developed becomes, more performance ability is needed. Considering the scale of this exercise, "professional skill for software development" (hereinafter described as "Skill") and "analytic ability for subject of software development" (hereinafter described as "Capability") are to be assumed as the factors which decide performance ability.

In addition, even thought it is better to include more alternative characteristics, the actually-obtained observed data is only 5 kinds described below.

6.2 Concentration of Alternative Characteristics Used for Analysis

Data used for analysis (values of alternative characteristics) are to be five variables, i.e. quantifiable skill (PGM1 and PGM2) and knowledge (M1, M2, and M3). Definitions thereof are as follows:
PGM1: The question to calculate the area of circle and ellipse by describing classes with overload.
PGM2: The question to rewrite the area calculating method for Rectangle set in Polygon to that for Triangle with override in order to calculate its area.
M1: The question about test coverage.
M2: The five questions below.
(1) A question of robustness analysis.

(2) A question to complete class illustration and to design classes.
(3) A question of class design which asks rules to transform the class of many-to-many multiplicity into the class of many-to-one.
(4) A question of class design to modify a class illustration with rules for transforming the class of many-to-many multiplicity into the class of many-to-one.
(5) A question to draw a sequence based on a class illustration.
M3: Knowledge for project management: a question about software life cycle model.

6.3 Data Collection

It was necessary to make teams within the period which is from commencement of the late course to start of the actual exercise. Therefore, usable data for team formation was limited. Concretely speaking, only two kinds of data were used: grades of final test of "software engineering" course conducted in the previous term and evaluation of students' programming skills obtained from the JAVA programming course conducted before this group exercise.

For reliability measurement of scale, reliability statistics α coefficient by Conbach is used. The reliability statistics is under 1, preferably 0.7 or 0.8 and above, and it should be 0.5 and above at least. Although the reliability statistics of the five variables were low as 0.355, a hypothetic model of CSA was formulated as using this value [14].

7 Verification Process of the Model and Result Thereof

7.1 Verification Process of the Model

For verification of the model, evaluation is conducted in favor of the indicator decided in view of the facts that sample size is small and the reliability of scale is low. At first the model converges and Chi-square test is carried out. Then fit index (total evaluation standard of the model) and evaluation of path coefficient (partial evaluation standard of the model) are conducted.

7.2 Verification Result of the Hypothetic Model

For Chi-square test, maximum-likelihood method to minimize divergence of measured values and theoretical values between dispersion and covariance of CSA.

The causal model between the roles and the alternative characteristics of Fig.2 is verified. This hypothetic model assures that optimization of team formation is dependant on two factors, and their latent variables are represented as "Skill" and "Capability".

In the analytic result of the model, value of the Chi-square (calculated value) is 0.347 and degrees of freedom is 4, which is smaller than the theoretical value, 0.987, in probability model; therefore, the hypothesis is not turned down (i.e. The model is correct.). The correlation coefficient between their latent variables, "Skill" and "Capability", is 0.00.

The goodness of fit shown in Table 4 is the model having highest fit ever because the value difference between 0.998 of GFI and 0.991 of AGFI is small, and value 22.347 of AIC is small(See Table5).

Table 4 Evaluation of model by fix index

model	GFI	AGFI	RMSEA	AIC
Model	0.998	0.991	0.000	22.347

Table5 Evaluation standard (referred AMOS7.0)

Index	reference value
GFI (Goodness of Fit Index）	GFI is less than or equal to 1. A value of 1 indicates a perfect fit.
AGFI (Adjusted Goodness of Fit Index）	GFI≧AGFI The AGFI is bounded above by one, which indicates a perfect fit. It is not, however, bounded below by zero, as the GFI is.
RMSEA (Root Mean Square Error of Approximation)	The smaller the RMSEA is the better. An RMSEA of zero indicates a perfect fit.
AIC (Akaike's Information Criterion)	This index compares and evaluates more than one model. A model with the smallest value of AIC is chosen.

Given this factor, the relational expression of two latent variables and alternative characteristics from Fig.2 is obtained as follows:

$$\left.\begin{array}{l} Skill=0.75\times PGM2+0.70\times PGM1+0.17\times M1 \\ Capability = 0.71\times M3+0.23\times M2+0.21\times M1 \end{array}\right\} \quad (4)$$

In addition, the t-test result at 5% of significance level of path coefficient, which is the partial evaluation of the model, showed insignificance in all the path coefficients.

8 Evaluation of Team Formation and Relevancy Thereof

8.1 Team Formation

Team formation is conducted with EtUDE/GO by the expression obtained from CSA. The role setting was firstly divided into four. However, with the obtained five alternative characteristics obtained, the assumption of this 4-role model provided inadequate solution even if the model was converged. Therefore, grouping is conducted with two latent variables, Skill and Capability.

8.2 Evaluation of Homogeneity of Team Formation with ANOVA

In order to make the best allocation for all the teams, reallocation is conducted even to the teams having no abstainee.

Each of Skill and Capability is verified its homogeneity with ANOVA (Analysis of variance).

Evaluation of ANOVA turns down null hypothesis if (P value < significance level set by experimenter) or (F border value, theoretical value of variance ratio when degrees of freedom is (m, n) at significance level α ≧ observed variance ratio).

Table 6 Result of ANOVA concerning Skill
(Single Factor)

Anova : Single Factor

SUMMARY

Groups	Count	Sum	Average	Variance
G1	4	-0.507	-0.127	3.296
G2	5	-2.226	-0.445	1.953
G3	5	-0.124	-0.025	2.258
G4	5	-1.131	-0.226	2.355
G5	5	-0.025	-0.005	3.203
G6	4	2.053	0.513	1.560
G7	4	1.683	0.421	0.387
G8	4	0.924	0.231	0.800
G9	4	1.829	0.457	1.484
G10	4	0.086	0.022	0.585
G11	4	0.451	0.113	1.160
G12	5	0.187	0.037	3.431
G14	5	-2.994	-0.599	1.546
G15	5	-0.206	-0.041	1.967

ANOVA Skill

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F cnt*
Between Groups	5.987	13	0.461	0.238	0.996	1.926
Within Groups	94.669	49	1.932			
Total	100.655	62				

In the analytic result of Skill (Table 6), the observed variance ratio is 0.238 and under 1.926 of the theoretical value of variant ratio when degrees of freedom is (13, 49) at 5% of significance level (F Criterion); therefore, the hypothesis as "There are differences among the groups" is turned down.

Table 7 Result of ANOVA
(Single Factor) concerning Capability

Anova : Single Factor

SUMMARY

Groups	Count	Sum	Average	Variance
G1	4	0.194	0.049	2.745
G2	5	-0.496	-0.099	0.585
G3	5	-0.564	-0.113	0.386
G4	5	-3.332	-0.666	0.651
G5	5	0.287	0.057	1.125
G6	4	1.214	0.304	0.161
G7	4	1.084	0.271	0.389
G8	4	-0.457	-0.114	1.255
G9	4	0.767	0.192	0.608
G10	4	0.443	0.111	0.198
G11	4	0.194	0.049	0.715
G12	5	-1.210	-0.242	0.645
G14	5	2.237	0.447	0.851
G15	5	-0.361	-0.072	0.458

ANOVA Capability

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F cnt*
Between Groups	4.600	13	0.354	0.468	0.932	1.926
Within Groups	37.016	49	0.755			
Total	41.615	62				

That the team formation by CSA satisfies the limiting conditions for new teams was confirmed with ANOVA.

9 Effectiveness Evaluation of the Optimization of Team Formation after the Course

9.1 How to Determine

After the exercise, the professors including TA evaluated all the students' contribution degree in their teams based on the presentation in the final rollout, final deliverables for system construction, and the log information obtained by EtUDE.

Grouping variable of objective variable in order to conduct discrimination analysis classified those who acquired high evaluation into Group 1 and the others into Group 0. The explanatory variables used for the evaluation are three which deducts 1 from four variables because qualitative data consist of four roles with dummy variable (0, 1).

9.2 Analytic Result of Team Formation

The result of the discrimination analysis by comparing role sharing and contribution degree in the teams showed that relative coefficient of canonical discrimination was 58.5% and 0.000 of significance probability was smaller than 0.05 of significance level α for Wilks lambda; therefore, it was recognizable that there was a difference between the two groups.

In the original grouped cases in Table 8, 77.2% of role sharing was correctly classified. In the grouped cases confirmed in crossover, 77.2% was also correctly classified.

This "Cross-validation" means the result of determination in the case of applying only one excluded example to the expression obtained except for the example. Therefore, the same result in both classification results means that an analysis on a new sample can be expected to bring similar discrimination result.

Table 8 Classification result of discrimination analysis

Classification result		assessment	prediction		Total
			0	1	
Original data	Freq.	0	31	1	32
		1	12	13	25
	%	0	96.9	3.1	100
		1	48.0	52.0	100
Cross-validation	Freq.	0	31	1	32
		1	12	13	25
	%	0	96.9	3.1	100
		1	46.2	53.8	100

Additionally, the cause that Group 1 has 52.0% of accuracy while 96.9% of Group 0 could be classified correctly is flexibility for the students to decide their roles. The division of roles in each team did not become what the authors expected, which led to prediction errors at the time of discrimination. In short, the students recognized that only the role of PM was important in a team but the others were not because of their difference in sense of value, which was turned out to be prediction errors seen in dispersion values concerning the allocation.

10. Conclusion

As a result of formulating the hypothetic model to analyze human errors influencing team formation in software development exercise, the relational expression between Skill and Capability as true factors and five alternative characteristics could be obtained. The optimal team formation scheme was generated by applying to the expression to EtUDE/GO. Based on the scheme, teams are made to conduct exercise course for the students.

After the 6-month course, the influences brought on the exercise by the optimization of the team formation were analyzed.

In the result, we could achieve the object that all the team would complete the exercise task with the optimization of the team formation.

We had expected that if the students having the ability to share the roles were allocated to each team, they would lead the team for achievement of exercise task. In the actual course, we confirmed that those students played their roles as the expectation and enabled all the teams to complete the task without delay. The problem to challenge next is the fact that we had to reorganize the teams in haste because some students canceled taking the course after the work for team formation with EtUDE/GO was done. Considering this, it is necessary to achieve "simplification work process for team formation". In addition, the analysis work for human factors in this experiment took much time to obtain the relational expression between the true factors and the alternative characteristics because the formulated hypothetic model was not converged easily due to small amount of data of the alternative characteristics which we could obtained. We aim at formulating a model taking advantage of more alternative characteristics with full preparation to clarify true factors before starting an exercise course of software development.

References:

[1] H. Hashiura, T. Kuwabara, Y. QIU, K. Yamashita, T. Ishikawa, K. Shirakawa, S. Komiya, "*A System for Supporting Group Exercise in Software Development with Facilities to Create an Optimal Plan of Grouping and Team Formation of Each Group*", The 7th WSEAS International Conference on Software Engineering, Parallel and Distributed Systems,2008.

[2] Yuka Isozaki , Koutarou Yamashita, Tatsuya Ishikawa, Hiroaki Hashiura, Seiichi Komiya, "*Environment for Ultimate software Development Exercise(EtUDE),*" Proceedings of the 67th National Convention of IPSJ, vol.4, 2005, 4D-4. (In Japanese)

[3] Akinori Iwasaki, Isao Miyaji, Takayuki Oue, "*Comparison between Exact Solution and Solution by GA in the Learning Group Composition,*" IEICE technical report. Education technology, vol.102, no.330, ET2002-40, pp.47-52, 2002. (In Japanese)

[4] K. Chiken, A. Hazeyama, "*Awareness Support in Group-based Software Engineering Education System Proceedings of the 10th Asia-Pacific*

Software and Applications Conference (APSEC2003)," IEEE Computer Society Press, pp.280-289, 2003.

[5] A. Hazeyama, N. Sawabe, S. Komiya, "*Group Organization System for Software Engineering Group Learning with Genetic Algorithm*," IEICE Transactions on Information and Systems, Vol. E85-D, No. 4, pp.666-673, 2002.

[6] Juan A. Marin-Garcia, Jaime Lloret, "*Improving Teamwork with University Engineering Students. The Effect of an Assessment Method to Prevent Shirking.*", WSEAS TRANSACTIONS on ADVANCES in ENGINEERING EDUCATION, Issue 1, Volume 5, January 2008

[7] Y. kano, " *Does Structural Equation Modeling Outperform Traditional Factor Analysis, Analysis of Variance and Path Analysis?* " The Behaviormetric Society of Japan, Vol.29, No.2(20021225) pp. 138-159 , The Japanese Journal of Behavior metrics, 2002. (In Japanese)

[8] Sun Microsystems Inc., "*Java Technology*," http://java.sun.com/

[9] Sun Microsystems, Inc., "*JSR-000154 Java Servlet 2.4 Specification*," http://jcp.org/aboutJava/communityprocess/final/jsr154/

[10] Sun Microsystems, Inc., "*JSR-000152 Java Server Pages 2.0 Specification*," http://jcp.org/aboutJava/communityprocess/final/jsr152/

[11] Sun Microsystems, Inc., "*Java EE at a Glance, 2006*," http://java.sun.com/javaee/

[12] Rick Knowles, "*Winstone Servlet Container*," http://winstone.sourceforge.net/

[13] MySQL AB, "*MySQL Community Server*," http://www.mysql.com/

[14] Y. Kano, "*The number of structural equation model for a number of indicators should be?*" The Behaviormetric Society of Japan, The Japanese Journal of Behavior metrics , Vol.31, No.2(20040910)pp. 143, 2004. (In Japanese)