CASE 85

Evaluation of Programmer Ability in Software Production

Abstract: This experiment was conducted to evaluate two types of ability: software error finding and program production. As a first step, we asked each testee to debug a program that included intentional errors. Then we evaluate his or her ability on the basis of the number of errors corrected and those not corrected. Because the results generated from a third party cannot be used to confirm experimental results and it is difficult to set the levels of human conditions, enough people need to be provided for a confirmatory experiment.

1. Evaluation of Software Error-Finding Ability

We asked testees to debug a program that had intentional errors. The results were two types of output data (Table 1). The types of error found can be expressed by a 0/1 value. A mistake can occur when no error is judged as an error. It is essential to judge an error as an error and also to judge no error as no error. Since Genichi Taguchi has proposed a method using the standard SN ratio for this type of case, we attempted to use this method.

We show the calculation process of the standard SN ratio as below. We set the total number of lines to n, the number of correct lines to n_0, that of lines including errors to n_1, that of correct lines judged as correct to n_{00}, that of correct lines judged as incorrect to n_{01}, that of incorrect lines judged as correct to n_{10}, and that of incorrect lines judged as incorrect to n_{00}. The input/output results are shown in Table 2.

Now the fraction of mistakes is

$$p = \frac{n_{01}}{n_0} \tag{1}$$

$$q = \frac{n_{10}}{n_1} \tag{2}$$

Using these, we calculated the standard SN ratio, η_0:

$$p_0 = \frac{1}{1 + \sqrt{[(1/p) - 1][(1/q) - 1]}} \tag{3}$$

$$\eta = -10 \log \left[\frac{1}{(1 - 2p_0)^2} - 1 \right] \quad \text{dB} \tag{4}$$

Choosing an L_{12} orthogonal array, we picked the 11 factors shown in Table 3. As a programming language, we used C and selected six scientific and engineering students with no knowledge of C. The number programs to be debugged was approximately 30. As a program type, a logical calculation program was chosen.

Each factor is defined as follows:

- *A: instruction time:* time needed for training in the C language
- *B: instructional method:* lecture by an instructor or individual study via software
- *C: number of reviewers:* review by a single person or by two people
- *D: review time:* time for reviewing
- *E: checklist:* whether or not to use a checklist
- *F, G, I: individual ability:* aptitude test (general job aptitude test issued by the Labor Ministry)

Table 1
Results of software error-finding experiment

Input	Output: Judged as Correct	Output: Judged as Incorrect
Correct	0	×
Incorrect (bug)	×	0

examining testee's ability in the Japanese language, business ability, and intellectual ability, and allocation of the results to an L_{12} orthogonal array

- *H: target value:* because of errors included in a program intentionally, whether or not to inform testees of the number of errors to be found
- *J: work intermission:* whether or not to give the testee a 10-minute pause to do other work that has no relation to the review work
- *K: number of errors:* number of errors included intentionally in a program

Using the control factors above, we performed the following experiment. First, we provided 4-hour-long instruction in the C language to six testees to conduct the experiments of six combinations as numbers 1 to 7 of L_{12}. Then we provided another four hours of training to the same testees (in total, eight hours of instruction) and conducted an experiment based on combinations of numbers 7 to 12 in the L_{12} orthogonal array. However, because each testee was dealing with three characteristics at the same time, it was difficult to allocate them independently. So by using a sequential approximation method, we estimated and calibrated SN ratios.

Table 2
Input/output for two types of mistakes

Input	Output: 0 (Correct)	Output: 1 (Incorrect)	Total
0 (correct)	n_{00}	n_{01}	n_0
1 (incorrect)	n_{10}	n_{11}	n_1
Total	r_0	r_1	n

Table 3
Factors and levels for experiments on error finding

Factor		Level 1	Level 2
A:	instruction time (hours)	4	2
B:	instructional method	Lectured	Self-taught
C:	number of reviewers	1	2
D:	review time (min)	20	30
E:	checklist	Yes	No
F:	ability in Japanese language	High	Normal
G:	ability for office work	High	Normal
H:	target	Yes	No
I:	intellectual ability	High	Normal
J:	work intermission	Yes	No
K:	number of errors	3	6

As a result of comparing the result obtained from this experiment and our prior experience, we could see that the trends in almost all of the control factors were understood satisfactorily. For instance, longer instruction time gave better results, and a review by two persons was more favorable than a review by one person. Therefore, we believe that the analysis method in our study contributed to evaluating factors related to error-finding capability.

2. Reproducibility of Results for Software Error-Finding Capability

As mentioned before, there are two types of errors: inability to find errors intentionally included in a program, and mistakenly considering a correct code to be an error.

While quality engineering requires a confirmatory experiment for reproducibility in gain under

an optimal configuration, we decided that instead of observing reproducibility under an optimal configuration, the trends of response graphs should be confirmed. Using new testees and a new program, we performed an experiment similar to our previous one.

However, to improve experimental reliability, we reiterated this new experiment based on a new program and repeated it three times (R_1, R_2, and R_3). Because this was a confirmatory experiment of the earlier one, we again selected six testees, an L_{12} orthogonal array, and 11 control factors (Table 3).

Following the conditions noted above, we obtained each datum. Figure 1 superimposes the response graphs of the first experiment on the confirmatory experiment. While the factor effects of ability in the Japanese language, business ability, and instruction time are still questionable, for other factors we see sufficient reproducibility. One of the possible reasons of poor reproducibility for ability in the Japanese language and business ability is that the reliability of the commercial aptitude test used to measure the two abilities is questionable. However, since we obtained fairly good reproducibility by changing testees and program, the reproducibility in this experiment can be regarded as sufficient.

3. Evaluation of Programming Capability

The previous experiment was conducted as follows. We handed specifications and a flowchart to a testee and asked him or her to write a program. When the testee had completed a program that was satisfactory to him or her, he or she saved the program. Next, compiling this with a compiler for the first time, the testee debugged it by himself or herself. After this program was checked by an examiner, a program with almost no bugs was obtained as a final output. Then, by comparing the two programs, we recognized a difference in code between them as an error.

In addition, we evaluated the data using an SN ratio for each error type. First, we classified errors into each type and allocated all of the types to a new orthogonal array (outer factor). As level 1, "use of error" was selected, whereas "no use of error" was level 2. By assigning these outer factors with the control factors (inner factors), thereby obtaining a direct product layout, we analyzed interactions between the control factors and noise types. Consequently, we had a total of 144 (12×12) SN ratios. The errors were classified as follows:

- A': insufficient understanding of specs, or eventual lack of functions
- B': a change in a program caused through introduction of a new idea
- C': functions, but incomplete
- D': inability of grasping variables and keeping track of data changes
- E': errors in a specific area despite correct codes in other areas (with no program framework)

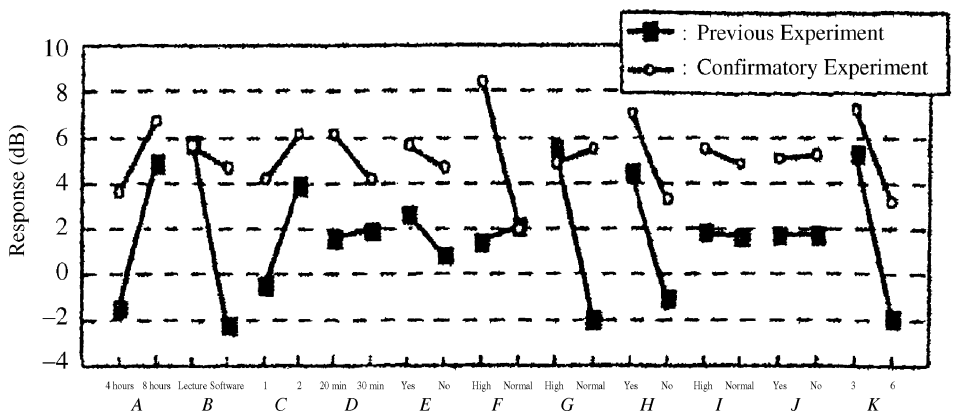

Figure 1
Response graphs for previous and confirmatory experiments

Table 4
Factors and levels for experiment and programming

Factor	Level 1	Level 2
A: type of program	Business	Mathematical
B: instructional method	Lectured	Self-taught
C: logical technique	F.C.	PAD
D: period before deadline (days)	3	2
E: coding guideline	Yes	No
F: ability in Japanese language	High	Normal
G: ability for office work	High	Normal
H: number of specs	Small	Large
I: intellectual ability	High	Normal
J: mathematical ability	High	Normal

- ❑ F': errors in a specific area despite correct codes in other areas (executable but incorrect program)
- ❑ G': correct idea but no knowledge of syntax
- ❑ H': inability to write codes
- ❑ I': addition of { }'s in accordance with an increase in program codes

The data are collected as follows:

Line number in program:	1	2	3	⋯	n
Representation of Correct Data:	y_1	y_2	y_3	⋯	y_n

where $y = 1$ for a correct line and $y = 0$ for incorrect line.

Table 5
ANOVA for experiment on programming

Factor	*f*	*S*	Factor	*f*	*S*
E	1	270.2	$G \times A'$	1	45.4
H	1	972.0	$H \times A'$	1	44.0
J	1	152.2	$I \times A'$	1	45.7
A'	1	124.4	$J \times A'$	1	44.8
B'	1	1461.1	$E \times B'$	1	319.7
$A \times A'$	1	58.6	$I \times B'$	1	115.4
$B \times A'$	1	51.8	$D \times C'$	1	93.8
$D \times A'$	1	67.9	$E \times C'$	1	69.4
e	76	767.7	$F \times C'$	1	59.2
			$G \times C'$	1	62.0
			$I \times C'$	1	73.9
			Total	95	4899.2

$V_e = 767.7/76 = 10.1$.

Table 6
Factors and levels for confirmatory experiment

		Level	
		1	2
A:	type of program	With objective	With objective
B:	instructional method	Lectured	Self-taught
C:	specs	With flowchart	With no flowchart
D:	period before deadline (days)	4	2
E:	test pattern	Yes	No
F:	ability in Japanese language	High	Normal
G:	ability for office work	High	Normal
H:	number of specs	Small	Large
I:	intellectual ability	High	Normal
J:	mathematical ability	High	Normal

We used the following calculation process. The fraction of the number of correct codes to that of lines, p, is computed as follows:

$$p = \frac{y_1 + y_2 + \cdots + y_n}{n} \tag{5}$$

An SN ratio for 0/1 data is expressed by a ratio of signal factor variation and error variation. To include additivity of measurement in the analysis, we expressed the SN ratio as a decibel value:

$$\eta = 10 \log \frac{S_p}{S_e} = -10 \log \left(\frac{1}{p} - 1 \right) \quad \text{dB} \tag{6}$$

An L_{12} orthogonal array was used for this experiment. As control factors, 10 types of factors were selected (Table 4). As an ability test, a commercial aptitude test was prepared. The C language is used and the program consisted of about 100 steps. As testees, six engineering students who had taken 10 hours of training in the C language were chosen. We tested them according to the levels allocated in the L_{12} orthogonal array.

The following is an explanation of the factors selected:

- *A: type of program:* type of program created by a testee
- *B: instructional method:* lecture on the C language from a teacher, or the use of instructional software
- *C: logical technique:* schematic of a program's flow
- *D: period before deadline:* requirement that program be handed in within two or three days after specifications are given
- *E: coding guideline:* material that states ways of thinking in programming
- *H: number of specs:* as long as the difficulty level of each program is not changed, minimal and

Table 7
Classification of errors in confirmatory experiment

	Level	
Factor	1	2
Logical error		
A′: mistake	Use	No use
B′: unknown	Use	No use
Syntax error		
C′: mistake	Use	No use
D′: unknown	Use	No use

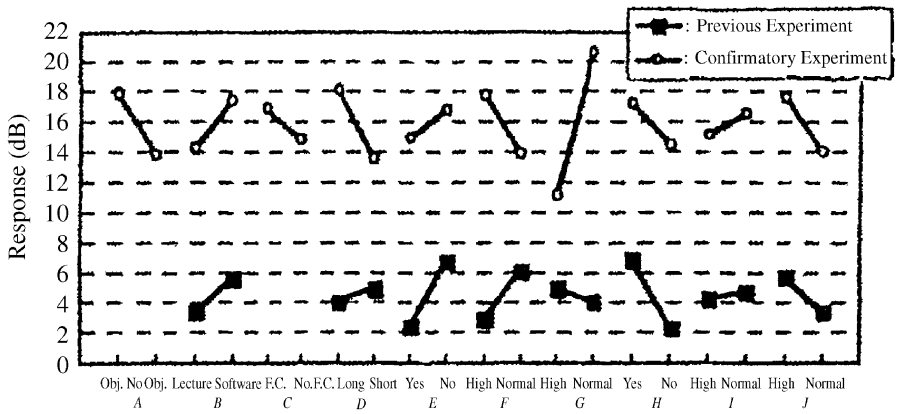

Figure 2
Comparison of response graphs between previous and confirmatory experiments

redundant specifications on which a testee was to write a program

- *F, G, I,* and *J: individual ability:* similar to abilities measured in the experiment on evaluation of error finding, plus mathematical ability

Since this experiment also tested six testees, using the same testees, we performed a separate experiment for the upper and lower halves of the orthogonal array. The results reveal that main effects are brought about by human abilities and number of specs. In addition, in terms of interactions between factors and errors, A', B', or the error of C' demonstrated a relatively large effect. That is, it was assumed that these errors were affected by individual differences (Tables 4 and 5).

Table 8
ANOVA for experiment on programming

Factor	*f*	*S*	Factor	*f*	*S*
A	1	425.5	$A \times A'$	1	655.6
B	1	266.2	$G \times A'$	1	327.1
D	1	474.5	$A \times B'$	1	820.0
F	1	369.4	$G \times B'$	1	398.5
G	1	2063.1	$A \times C'$	1	409.5
K	1	319.4	$A \times D'$	1	211.7
A'	1	1952.3	$C \times D'$	1	229.0
D'	1	911.2	$D \times D'$	1	382.9
e	77	3061.8	$H \times D'$	1	417.0
			$K \times D'$	1	215.7
			Total	95	13,910.3

$V_e = 3061.8/77 = 39.8$.

4. Reproducibility of Results for Programming Capability Evaluation

This experiment was also performed under almost the same conditions. The only difference was addition of repetitions (R_1 and R_2) and a change in some control factors. Because of the repetitions added, considering the increased workload on the testees, we prepared two 50-line programs in this experiment instead of one 100-line program.

We describe the changed factors as follows (see Table 6):

- *A: type of program.* Because of using a different program, we could not follow the classification in the previous experiment. Rather, we focused on whether or not the objective of a program was clarified.
- *C: specs.* Since we did not have enough time to instruct the testees in two types of flow-

Table 9
Gain obtained from experiments on error finding and programming

	Experiment on Error Finding		Experiment on Programming	
	Previous	Confirmatory	Previous	Confirmatory
All factors				
Optimal	22.6	17.6	13.3	31.0
Worst	−19.1	−7.3	0.0	0.5
Gain	41.6	24.9	13.4	30.5
All factors except Japanese language, office work, and time				
Optimal	18.4	13.0	12.9	22.2
Worst	−14.9	−2.7	0.4	9.4
Gain	12.6	19.0	12.6	12.8

charts, in this experiment we considered whether a flowchart was included in the specs.

- *D: period before deadline.* Because there is little difference between two and three days, we changed the periods to two and four days so that the difference would be greater.
- *E: test pattern.* The existence and nonexistence of a test pattern were compared.

Since there were approximately 50 lines in one program in the confirmatory experiment, we cannot classify the errors due to the small number as we have done in the preceding experiment. Additionally, because a small number of lines mitigates differences among individuals, we proceeded with the analysis following the categorization of errors shown in Table 7.

Figure 2 illustrates the response graphs. For the factors in the preceding experiment, we indicated only the factors that were the same as, or similar to, the corresponding ones in the confirmatory experiment. As a result, we can see that many of the factors have good reproducibility. On the other hand, it is quite reasonable that ability in the Japanese language, business ability, and period before deadline (time) still have poor reproducibility because all of them show the same conclusions in the experiment on error-finding capability. Table 8 lists the ANOVA for the experiment, and Figure 2 compares the response graphs.

5. Confirmation of Gains in Experiments on Error Finding and Programming

Table 9 shows calculation of the gains. While the upper half represents a normal calculation of gain, the lower indicates a computation excluding ability of Japanese language, business ability, and time, all of which had poor reproducibility. Because of no current configuration in our study, we defined a difference between SN ratios under the optimal and worst configurations. When using all the control factors, the reproducibility was extremely poor, but we obtained fairly good reproducibility when some factors were excluded.

According to the two experiments discussed thus far, we can see that effects caused by interactions need to be considered in an analysis because of significant interactions among control factors (differences among individuals) in the case of dealing with a 100-line program, whereas we can perform an analysis with no consideration of interactions when a 30- to-50-line program is used. On the other hand, since the commercial aptitude test used for evaluating human ability factors is problematic in terms of reliability, we need to seek a substitute.

Yet since we obtained sufficient reproducibility in an operational process including human factors, it would be reasonable to say that our analysis method was effective for the evaluation of programming capability.

Reference

Kei Takada, Muneo Takahashi, Narushi Yamanouchi, and Hiroshi Yano, 1998. The study of reappearance of experiment in evaluation of programmer's ability of software. *Quality Engineering*, Vol. 6, No. 1, pp. 39–47.

This case study is contributed by Kei Takada, Muneo Takahashi, Narushi Yamanouchi, and Hiroshi Yano.