

Analysis of fault generation caused by stress during software development

Tsuneo FURUYAMA, Yoshio ARAI, and Kazuhiko IIO

NTT Software Laboratories

9-11, Midori-cho 3-chome, Musashino-shi, Tokyo 180, Japan

Tel. +81-422-59-2540 Fax. +81-422-59-3712 E-mail:

furuyama@slab.ntt.jp

Abstract

The effects of stress on fault generation in both structured and functional design methodologies were quantitatively determined through a controlled experiment. Two teams developed the same software program under the same stressed conditions, except for design methodologies. The degree of stress was measured by inner metrics we proposed (Furuyama, 1994-b). The results of the analysis show that (1) the generation rate of faults caused by mental stress of the team who developed a software program using functional design methodology was higher than that of the other team, which used structured design methodology; (2) among the faults caused by human nature, many seemed to be correlated to stress at higher stress levels, so that faults caused by stress were considered to be generated by developers much more than they reported them; and (3) physical stress could generate faults at a higher rate than mental stress when it appeared even for a short period, independent of design methodology.

Keywords

Quality improvement, Fault analysis, Quantitative quality evaluation, Mental stress, Physical stress, Stress metrics, Design methodology, Software engineering

1 INTRODUCTION

Estimating and improving reliability is one of the most important areas in software development management. In the early history of software development, techniques for detecting and eliminating faults during the test phase and estimating the remaining faults were important. Later, the importance of estimating and improving reliability during the design phase was recognized. Design review techniques, such as formal inspection, were developed and became widely used in actual software development.

The purpose of these techniques, however, is to detect and eliminate existing faults and to estimate the remaining faults. A more effective approach to improving reliability would be to prevent faults during the early stages of development. One proposed way to reduce faults is to decrease software complexity, which is a significant factor in fault generation (Basili, 1984 and Takahashi, 1984). In this approach, however, human factors are generally not explicitly considered because it is difficult to measure and analyze human behavior.

Human factors have been discussed in the area of mechanical engineering for a long time as one of the major influences on system reliability. However, most research into human error has focused on operational aspects (Reason, 1987 and Lee, 1988). Among them, Rasmussen et al., in their distinguished work, used cognitive science techniques in their model and produced a comprehensive framework to describe the mechanisms behind the cause of human errors (Rasmussen, 1987).

In the area of software development, however, human factors have been little discussed, despite the fact that human factors seem to affect software reliability, and it is therefore important to develop methods that minimize the impact of human factors to improve reliability.

Nakajo and Kume, in their systematic description, stated briefly that fault generation is the result of human factors, that is, human misunderstandings as well as the circumstances in which faults are likely occur and the generation of faults is related to the system interface or system functions. However, their fault generation model does not clearly define such factors as the mental state of the programmer, making it difficult to draw conclusions regarding the prevention of faults.

One of the reasons that clarifying the effect of human factors during software development is so difficult is that they are concerned only with human thought, which, unlike mechanical operations, is invisible and hence difficult to measure.

To clarify the effect of human factors during software development, we first proposed a fault generation model (Figure 1) for software development taking human factors into consideration, based on our interviews with developers (Furuyama, 1994-a). Then, we estimated that about a third of the generated faults could have been avoided by identifying the factors causing them and removing those factors. Our finding was that mental stress greatly affects fault generation.

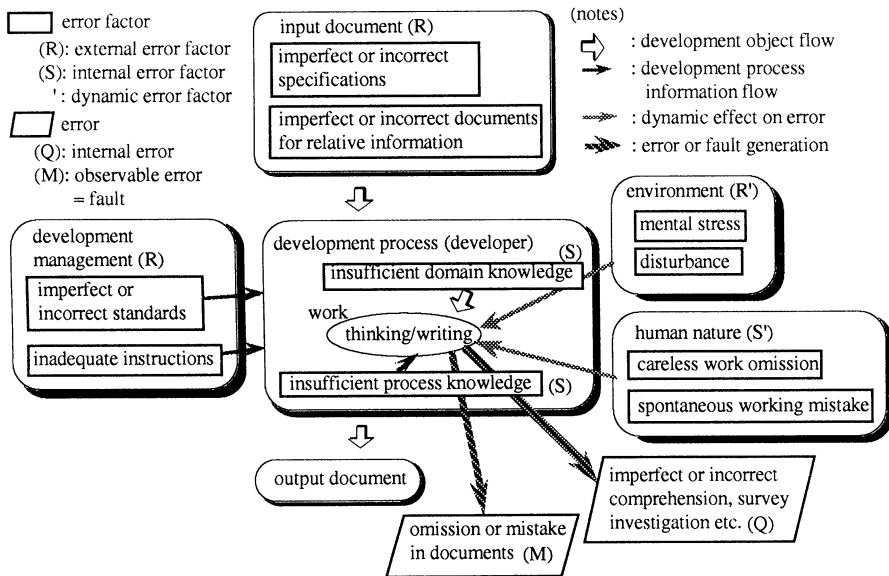


Figure 1 Abstract Model of Fault Generation (modified from Furuyama, 1994-a).

Software developer stress factors such as short development times were earlier researched by Fujigaki, based on questionnaires sent to software specialists and her analysis of developer dialogs during the design phase (Fujigaki, 1992). However, her data included many factors that could not be clearly separated, making it difficult to isolate the effects of stress on fault generation.

Next, we conducted controlled development experiments to isolate and measure quantitatively the mental stress effect on fault generation, developing metrics that measure mental stress easily and objectively in actual software development. Two teams developed software based on the same specifications under nearly the same circumstances, except for mental stress. The team skill levels were nearly equal. Our 'inner stress metrics' system was used by the developers to report their mental conditions in predefined formats. These metrics were collected every day. The results of our analysis showed that this metric system is effective for predicting how many faults might be generated according to the mental stress level (Furuyama, 1994-b).

Takahashi, et. al. compared the data obtained from two teams developing the same software but using different design methodologies, and found that the software reliability of the team using structured design methodology was higher than that of the team using functional design methodology (Takahashi, 1995). However, in their experiment, no consideration was given to the development environment, especially to the mental and physical stress which usually occur in an actual development environment.

To clarify further details of the effect of stress, not only mental but also physical, on software development -- in particular, the difference between design methodologies -- we again carefully designed the controlled experiment with the aim of quantitatively determining the difference of stress effect on fault generation between design methodologies.

2 EXPERIMENT

In our experiment, two teams developed the same software under the same mental stress conditions, except for design methodologies. During the development process, the degree of the developers' mental and physical stress, the number of generated faults and their characteristics, and general working condition data were collected every day. The experiments were designed so that such factors as developer skill and work environment were nearly the same for both teams.

2.1 Outline of development

System developed

The system each team developed was a 'library management system' to support such functions as searching, lending, returning, and registering libraries. The program size was about 5 KLOC in C language.

Design methodologies

Two different methodologies were examined, functional design and structured design. Table 1 compares the two methodologies. Their main differences are as follows.

(a) In the functional design phase, the functional design team was required to make only functional specifications narratively described, while the structured design team was additionally required to produce more rigorously structured documents, such as data flow and data structure diagrams. These additional documents required much efforts though their volume was smaller than that of the functional specifications.

(b) In the detail design phase, the functional design team was required to make only a module structure chart, while the structured design team was additionally required to produce a

structure chart showing data flow between modules. Both teams were required to produce a Hierarchical Compact (HCP) chart.

Team organization

Each team consisted of two developers, who were instructed to design the system collaboratively, except for the documentation, within each team. To minimize the differences between both teams in terms of development ability and skill, the developer with the highest development skill and the developer with the lowest were assigned to one team (Team A), which used functional design methodology, and the two intermediate-skill developers were assigned to the other (Team B), which used structured design methodology.

Though both teams had experience in functional design methodology, Team B had no experience in structured design methodology. Therefore, Team B received training in that methodology.

Table 1 Comparison of products between the two methodologies

Phase	Functional Design Methodology	Structured Design Methodology
FD	Functional design document	Functional design document
	-	Data flow diagram
	-	Data structure diagram
DD	Module structure chart	Module structure chart with data flow between modules
	HCP chart	HCP chart

Work environment

To equalize the physical development circumstances and to avoid team interactions, the teams were placed in separate rooms with nearly the same space, lighting, and noise level. The developers were forbidden to speak with the members of other teams about the development work.

Table 2 Development efforts

Phase	Team A (mon-hours)	Team B (mon-hours)
FD	119	192
FD review	98	70
DD	250	321
DD review	100	59
M/DB	180	158
Total	747	800

Development schedule

The total development time was about three months. Based on the report that design time between the two design methodologies varies (Oka, 1993), we determined the standard design schedules of the two teams such that the schedule of Team B was longer than that of Team A by 50% in the functional design (FD) phase and by 30% in the detail design (DD) phase respectively. Table 2 shows the actual development efforts of both teams.

2.2 Creation and measurement of mental stress

Creation of mental stress

During the software development, stress factors in terms of 'work contents' such as schedule pressure and too much work are the major factors (Fujigaki, 1992, Furuyama, 1994-a and Furuyama, 1994-b). To analyze the effects of these factors in detail, the following stresses were given to both teams during both the FD phase and DD phase:

- Design time was reduced by 30% less than the standard schedule, and
- Specification changes were imposed three times in both design phases.

Metrics for mental stress

Stress is generally measured physiologically (by electroencephalograph, electrocardiogram, heart rate, respiration volume), biochemically (by catechol amine in urine or in blood), and psychologically (by subjects' answers to questions about stress-related conditions). Among these measurement metrics, psychological metrics are effective and easily applied to actual software development. In particular, the 'schedule pressure' metric was the most effective for detecting the degree of mental stress, and the 'workload' metric had the highest correlation coefficient to the number of stress-caused faults (Furuyama, 1994-b). Based on these results, these metrics were selected for measuring the degree of mental stress of all developers in this experiment.

Metrics for physical stress

Besides mental stress, another type of stress directly affects developers' intellectual conditions and therefore leads to fault generation. For example, catching a cold lowers developer working efficiency. To analyze the effect of this type of stress, a 'physical condition' metric was also selected to measure physical stress. Since other types of physical conditions such as fatigue caused by overwork seemed to be correlated to mental stress, we did not identify them as separate physical metrics.

Measurement of stress

Metrics values from -6 to +6 were recorded by the developers themselves at the end of their work every day in a questionnaire (Figure 2), though the scale endpoints in the previous experiment were at -5 to +5.

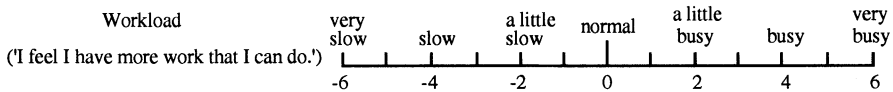


Figure 2 Example of stress questionnaire item.

2.3 Recording works and fault report

All developers recorded their work in detail in daily work diaries so that all faults detected could be easily tracked and the moment and the causes of fault generation could be accurately identified from the fault phenomena.

The faults report recorded fault occurrences, the error factors causing the faults, and the fault generation date. It also recorded the stress metric values so that the relationships between the error factors and the stress metrics could be easily discerned. Faults detected during design were counted, as well as those detected during design review and testing. Spelling mistakes were excluded.

3 RESULTS

3.1 Degree of stress

Figure 3 shows the degree of mental and physical stress of all developers measured with 'schedule pressure,' 'workload,' and 'physical condition' metrics. All metrics values were equal to zero when developers felt no stress, such as a few days before starting this experimental project. We assumed that the same metric values among developers means that they were loaded with the same degree of stress.

Table 3 shows the result of the F-test of mean stress level between the two teams.

Analysis of the results as shown in Figure 3 and Table 3 leads to the following conclusions.

- (a) On the whole, both teams suffered from the same level of stress during the FD phase.
- (b) Team A suffered from much more stress than Team B during the DD phase, but if developer B2, whose stress level was significantly lower than others, is excluded, stress levels of both teams become equal.
- (c) Only developer A2 and developer B2 were suffering from poor health during both the FD and the DD phase.

Table 4 shows the correlation coefficients between all three metrics values. From Table 4, the following conclusion is also obtained.

- (d) Two mental stress metric values were highly correlated to each other, while physical metric values had little correlation to either of the mental stress metrics values, as we assumed in 2.2.2.

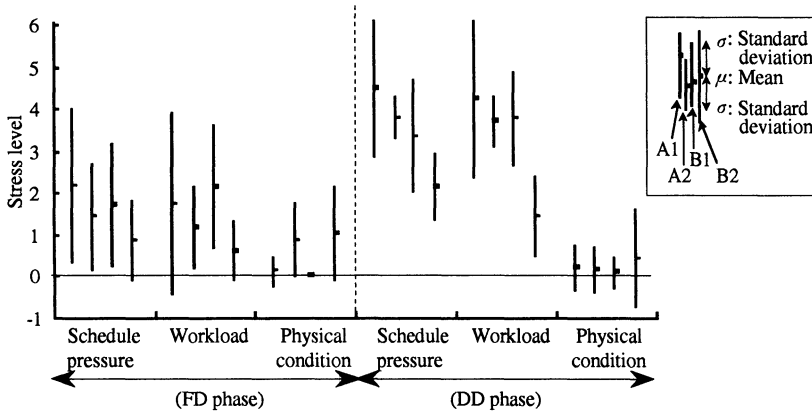


Figure 3 Stress metrics values.

Table 3 F-test of mental stress metrics between Team A and Team B

Phase	Stress Metrics	Team A		Team B			d.f.	t	
		Mean	Variance	Points	Mean	Variance			Points
FD	Schedule pressure	1.81	2.544	32	1.35	1.721	52	56.3	1.39
	Workload	1.47	2.838	32	1.38	1.888	52	55.9	0.24
DD	Schedule pressure	4.16	1.547	62	2.75	1.534	68	126.8	6.48*
	Workload	4.03	1.999	62	2.62	2.449	68	128.0	5.42*

(*) 1% significance

Table 4 Correlation coefficient between three metrics

Developer	Schedule pressure and Workload		Schedule pressure and Physical condition		Workload and Physical condition	
	FD phase	DD phase	FD phase	DD phase	FD phase	DD phase
A1	0.98*	0.92*	--	--	--	--
A2	0.95*	0.86*	--	--	--	--
B1	0.85*	0.81*	--	0.36**	--	--
B2	0.77*	0.72*	--	0.71*	0.48**	0.62*

(*) 1% significance, (**) 5% significance

3.2 Fault classification

Generated faults were classified into those caused by static factors and those caused by dynamic factors due to human error, or 'dynamic error factors' for short, as shown Figure 1, based on the fault reports in which error factors of all faults had been determined by developers. The former consists of faults caused by inadequate input documents, such as imperfect or incorrect input specifications (*what* to be developed); by inadequate development management, such as imperfect or incorrect work standards (*how* to develop); and by developers' lack of knowledge needing supplementation when designing software. The dynamic error factors consist of faults caused by stress, and faults caused by human nature.

The faults were classified according to the following priority: faults caused by static factors > faults caused by stress > faults caused by human nature. For example, if a fault apparently resulted from incorrect specifications, it was classified into 'faults caused by static factors,' regardless of the level of stress conditions. Similarly, if a fault was apparently caused by strong stress conditions, it was classified into 'faults caused by stress,' even if the developer overlooked a necessary part of his work because of his overconfidence -- a human nature error. Therefore, 'faults caused by human nature' can be said to be those whose error factors are not necessarily clear.

3.3 Generated faults

Table 5 shows the distribution of the fault generation patterns classified according to the two teams (two design methodologies), development phases, and fault-causing error factors. Since only four faults were considered to be caused by physical conditions, we decided to analyze them separately in detail (discussed later in 4.3) and excluded them from Table 5.

Table 5 Distribution of fault generation patterns

Error Factors	FD phase			DD phase		
	Team A	Team B	Total	Team A	Team B	Total
Static	14	15	29	8	13	21
Stress	20	14	34	85	42	127
Human nature	22	15	37	10	23	33
total	56	44	100	103	78	181

General trends of the generated faults were as follows.

Difference between teams and between design phases

- (a) Both teams generated twice as many faults during the DD phase as during the FD phase.
- (b) The number of faults generated by Team B (the structured design team) was less than that of Team A (the functional design team) by about 30% during both the FD and the DD phase (A : B = 53 : 40 during the FD phase and 103 : 78 during the DD phase). This will be discussed in detail in 4.1.

Faults caused by static factors

- (c) No statistical significance was found between the two design methodologies in terms of faults caused by static factors generated during the FD phase (A : B = 14 : 15). Though there is little statistical significance between them (8 : 13) in the kind of faults during the DD phase, the main reason that Team B generated more faults than Team A was their inexperience with the HCP chart used during the DD phase compared to Team A, and the HCP chart was considered to be a problem specific to Team B.

(d) In both teams, most of the faults caused by static factors were due to lack of domain knowledge in the FD phase (A : B = 11/14 : 11/15) and lack of design knowledge in the DD phase (5/8 : 10/13). This is because domain knowledge is an important element in the functional design, and design knowledge is an important element in the detail design phase.

Faults caused by dynamic error factors

(e) Faults caused by stress accounted for a high proportion of all generated faults. More than a third of the faults generated during the FD phase (A: 36%, 20 faults; B: 32%, 14 faults) and more than a half of all faults generated during the DD phase (A: 82%, 85 faults; B: 54%, 42 faults) were due to mental stress.

(f) The number of faults caused by human nature was nearly equal to that caused by mental stress in the FD phase, and lower than that in the DD phase, which was a stressful development environment.

The results of (e) and (f) were consistent those of our previous report (Furuyama, 1994-b): stressful conditions over time shifted some of the developers' attribution of faults from the human nature category to the stress category.

4 ANALYSIS

Figure 4 shows a model of the relationships between stress, mental and physical conditions, fault generation, and their measurement. Physical stress was measured in addition to mental stress. Both stresses are considered to seriously affect human intellectual behavior.

Using this model, the collected data were analyzed from the following viewpoints.

- Are there any differences, in terms of fault generation, between functional design methodology and structured design methodology in stressful circumstances?
- In addition to mental stress metrics, is a 'physical condition' metric related to fault generation?

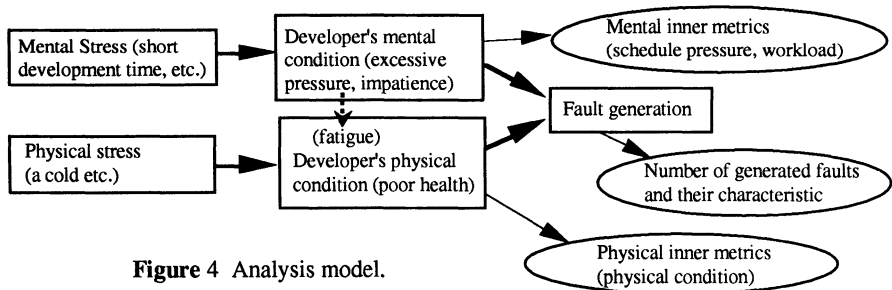


Figure 4 Analysis model.

4.1 Mental stress effect on fault generation between two design methodologies

Calculation of fault generation rate

Since there was occasional overtime work and working time varied widely every half-day (morning and afternoon), to merely count and compare the raw number of faults generated in every-half day as a unit for measuring stress level was inadequate. Therefore, we accumulated working time for calculating efforts and generated faults level by level, and compared the fault generation rate obtained by dividing accumulated faults by accumulated working time level by level (Table 6).

'Workload' metric values were used as representative to classify the faults level by level, because both mental stress metric values were correlated to each other as shown in Table 4, and 'workload' had a little higher correlation to fault generation than 'schedule pressure' (Furuyama, 1994-b).

There were two types of fault generation rate each level: the faults per actual effort in man-hours, and the normalized rate by total design effort. These differed both between the two design methodologies and between the FD and the DD phases. The latter type was derived from the concept that the fault generation rate should be normalized by the effort needed to complete the same function, such as the Function Point concept. In other words, the fault generation rate should be calculated as faults generated per working steps, to build up a unit function. We adopted the normalized rate to compensate for the difference in efforts between the two teams. Speaking more concretely, each fault generation rate of Team A was calculated by dividing the number of faults by the actual effort themselves, and that of Team B was compensated by multiplication of the ratio of total efforts of Team B to that of Team A after dividing the number of faults by actual efforts. For example, in the FD phase, every fault generation rate at each level for Team B in the FD phase was compensated by the value 191.5/119 obtained from Table 2. (Hereafter, 'normalized fault generation rate' is expressed as 'fault generation rate' for short).

Table 6 Faults caused by dynamic error factors and the generation rate classified by mental stress level

Phase	Team A			Team B			
	Stress level (Work-load)	Accumulated effort (man-hours)	Generated faults		Accumulated effort (man-hours)	Generated faults	
			Caused by stress	Caused by human nature		Caused by stress	Caused by human nature
FD	0	41.25	2 (0.05)	6 (0.15)	49.75	0 (0.00)	3 (0.10)
	1	24.75	1 (0.04)	2 (0.08)	66.50	1 (0.02)	5 (0.12)
	2	16.50	1 (0.06)	2 (0.12)	31.25	2 (0.10)	0 (0.00)
	3	9.50	2 (0.21)	6 (0.63)	0.00	--	--
	4	19.00	9 (0.47)	6 (0.32)	44.00	11 (0.40)	7 (0.26)
	5	0.00	--	--	0.00	--	--
	6	8.00	5 (0.63)	0 (0.00)	0.00	--	--
Total	119.00	20 (0.17)	22 (0.18)	191.50	14 (0.07)	15 (0.08)	
DD	0	4.25	0 (0.00)	0 (0.00)	21.75	0 (0.00)	1 (0.06)
	1	2.00	0 (0.00)	0 (0.00)	54.75	11 (0.26)	7 (0.16)
	2	28.00	7 (0.25)	0 (0.00)	83.75	15 (0.23)	5 (0.08)
	3	18.00	1 (0.06)	0 (0.00)	31.75	1.5 (0.06)	1 (0.04)
	4	144.50	65 (0.45)	6 (0.04)	107.25	13.5 (0.16)	7 (0.08)
	5	0.00	--	--	8.75	0 (0.00)	0 (0.00)
	6	53.00	12 (0.23)	4 (0.08)	13.25	1 (0.10)	2 (0.19)
Total	249.75	85 (0.34)	10 (0.04)	321.25	42 (0.17)	23 (0.09)	

(): fault generation rate (faults/man-hour); Team B is normalized by the ratio of total effort of Team B to that of Team A.

Comparison of fault generation rate between two teams

Our analysis led to the following conclusions.

(a) Faults caused by mental stress (hereafter sometimes expressed merely as 'stress') of Team A in the FD phase (0.17 faults per man-hour) were much more numerous than those of Team B

(0.07 faults per normalized man-hour) even after normalization, as shown Table 6. To clarify the differences of fault generation between the two teams in detail, we investigated faults and generation rate at each mental stress level. The difference was found to be due to five faults generated at level 6 of Team A as shown in Table 6, while Team B did not attain stress level 6 at all and therefore had no faults at that level. If we exclude these faults at level 6, **there is no difference in normalized fault generation rate at all between the two teams in FD phase**, in that the normalized fault generation rate of Team A was a little higher than that of Team B at stress level 1, a little lower at stress level 2, and nearly equal at stress level 4.

(b) **In the DD phase, faults caused by stress per actual and per total design time of Team A were both much more numerous than those of Team B.** The overall difference was derived from the extreme difference of generated faults at level 4 between the teams.

(c) In the FD phase, no significant differences of normalized fault generation rate for faults caused by human nature were found between the teams, except for the difference at stress level 3, where the fault generation rate of Team A was very high while there was no data at the level of Team B, thus making comparison impossible. The difference at level 3 is discussed in detail in 4.2.

(d) In the DD phase, total faults caused by human nature of Team A (10) were less than those of Team B (27). This seemed to derive from the shift from the human nature category to the stress category, as described earlier.

(e) **The total number of faults caused by dynamic error factors of Team A was much greater than those of Team B**, even considering the difference of stress level between the teams. This means that using structured design methodology rather than functional design methodology can improve reliability not only in non-stressed circumstances, as reported by Takahashi et. al (Takahashi, 1995), but also in stressed circumstances.

Stress level and fault generation rate

Table 7 shows the correlation coefficients between stress level and fault generation rate.

Table 7 Correlation coefficients between stress level and generated faults

Phase	Team	Faults caused	
		by stress	by human nature
FD	A	0.94*	0.01
	B	0.96**	0.58
DD	A	0.63	0.89**
	B	-0.22	0.09

(*) 1% significance, (**) 5% significance

B had no correlation to stress level.

(f) In the FD phase, faults caused by stress increased proportionate to stress level. A high correlation was found between stress level and generated faults caused by mental stress (Table 7). The regression coefficients were nearly equal to each other: 0.109 and 0.104 respectively (Table 8 and Figure 5).

(g) In the DD phase, Team A's generation rate of faults caused by stress was correlated to stress level, though it was not statistically significant, while that of Team

Table 8 Analysis of variance of stress level and faults caused by mental stress during FD phase

Team	Factor	d.f.	Sum of square	F-value	Contribution rate	Regression coefficient	Constant
A	Regression	1	0.2783	32.6*	0.891	0.109	-0.048
	Residual	4	0.0341				
B	Regression	1	0.0952	24.6**	0.925	0.104	-0.050
	Residual	2	0.0077				

(*) 1% significant, (**) 5% significant

(h) In the FD phase, there seemed to be correlations between stress level and generated faults caused by human nature in both teams; Team A could have had a higher correlation coefficient if the fault generation rate at level 3 whose value was much higher than others were ignored, and Team B had correlation coefficient 0.58.

(i) In the DD phase, there was significant correlation between stress level and faults caused by human nature for Team A, while there was no correlation in Team B as to faults caused by stress. Detail analysis will be described below in 4.2.

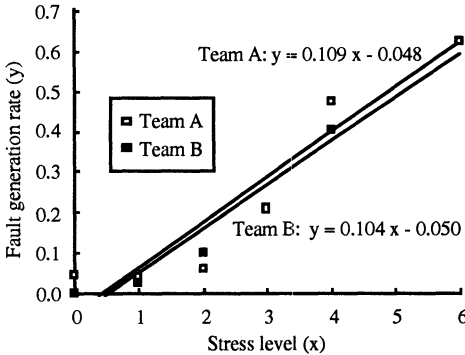


Figure 5 Generation rate of faults caused by mental stress in the FD phase.

Discussion

An unexpected result was that in the FD phase, there was no difference in the normalized generation rate of faults caused by stress between the two methodologies ((a) and (f)). One reason for this result would be, as described above, that faults may be generated according to the amount of work done to accomplish a unit function, because in other respects the working or thinking steps would be similar within a family of design methodologies, as in the case of our experiment, even if the documents to be made were different.

The better effect of using structured design methodology to reduce faults caused by stress appeared, instead, in the

DD phase((b)). The reason may be due to 'structured and sufficient input documents' made during the FD phase.

As a whole, faults caused by dynamic error factors of Team B were fewer than those of Team A, slightly in the FD phase and appreciably in DD phase, and consequently throughout all the design phase. Therefore, it becomes clear that **structured design methodology is much more suitable for developing reliable software, even in a stressed environment, than functional design methodology.** One significant reason that structured design methodology is highly tolerant of a stressed environment may be that developers using it design software in a more deliberate, orderly way and write the results in comprehensive documents in an earlier design phase than is done in functional design methodology.

For the increasing rate of fault generation in the FD phase as the stress level increases, the result of 0.109 per hour was consistent with the previous result, 0.86 per (8-hour) day, which was obtained by a team also using functional design methodology (Furuyama, 1994-b), even considering the difference of maximum scales of stress metrics between +6 and +5. This means **these controlled experiments can have consistent results** even though they deal with human factors heretofore considered too difficult to investigate.

The reason that the fault generation rate of both teams was less correlated to stress in the DD phase than in the FD phase seemed to be that the work in the DD phase was more procedural than that in the FD phase. In particular, Team B had no correlation to stress level, though they generated many faults caused by stress. This was due to the input documents used in the DD phase that were written rigorously in the FD phase.

Though Team B spent more time in the design phase (in particular, in the FD phase nearly twice as long), compared to Team A, their total effort was nearly equal as shown Table 2. However, in this experiment, almost all of the faults were detected at the end of the DD phase by experimenters and hence correcting efforts for remaining faults were almost eliminated. If the fault detection rate of both teams by design review had been equivalent, the remaining undetected faults of Team A would have been much greater than those of Team B, and more

efforts would have been needed than was the case in this experiment. This means that it is important to design software in a structured way, to reduce fault and development effort.

4.2 Indirect stress effect on fault generation caused by human nature

By definition, faults caused by human nature are to be considered as not dependent on stress. In fact, they tend to decrease as a whole as stress increases, because the period time when developers are conscious of schedule pressure may decrease. However, as described in 4.1, it was observed that most of generation rate of faults caused by human nature seemed to have correlation to stress level. In this section, we analyze the reason in detail.

Table 9 shows the classification of fault generation according to whether two types of faults were generated alone or together when stress was loaded on the developers. If the generation of both types of faults are independent, co-generation factor η , as defined by the following expression:

$$\eta = (P1/P2)/(P3/P4) = (P1*P4)/(P2*P3) , \tag{1}$$

is equal to one, where P1 is the probability that neither faults caused by stress nor by human nature are generated, P2 and P3 are the probabilities that only one type of fault is generated, and P4 is the probability that both types of faults are generated together in a unit time.

All half days were assumed to be equal. In each of the tabulations in Table 9, P1 is the lower right, P2 and P3 are the upper right and lower left items, and P4 is the upper left item respectively. Table 9 also shows that η was mostly much greater than one. This means there was strong dependence between the two types of fault generation. In other words, **in their overconfidence and not fully aware of their level of stress, developers might have misattributed stress-caused faults to human nature instead.**

Table 9 Co-generation of faults caused by stress vs. those caused by human nature

<u>(a) Team A in the FD phase</u>			<u>(b) Team B in the FD phase</u>		
<u>Faults caused by human nature</u>	<u>Faults caused by stress</u>		<u>Faults caused by human nature</u>	<u>Faults caused by stress</u>	
	<u>Generated</u>	<u>Non-generated</u>		<u>Generated</u>	<u>Non-generated</u>
Generated	16 (8/14)	6.75 (--/2)	Generated	21.5 (12/9)	9 (--/13)
Non-generated	20 (10/--)	35	Non-generated	8 (2/--)	103.25
$(\eta=4.2)$		(hours)	$(\eta=30.1)$		(hours)
<u>(c) Team A in the DD phase</u>			<u>(d) Team B in the DD phase</u>		
<u>Faults caused by human nature</u>	<u>Faults caused by stress</u>		<u>Faults caused by human nature</u>	<u>Faults caused by stress</u>	
	<u>Generated</u>	<u>Non-generated</u>		<u>Generated</u>	<u>Non-generated</u>
Generated	24.25 (8/9)	5.25 (--/1)	Generated	73.75 (27/15)	25 (--/7)
Non-generated	118.75 (77/--)	97.25	Non-generated	79.75 (15/--)	121
$(\eta=3.8)$		(hours)	$(\eta=4.5)$		(hours)

Note: Data from stress level zero is excluded

Legend: (faults caused by stress/faults caused by human nature)

This assumption was reinforced by the following example. Since so many faults caused by human nature in Team A were generated at level 3 in the FD phase, as described in 4.1 (c), we analyzed this in detail and found that developer A2 had generated two faults caused by stress followed by six faults caused by human nature in a half-day while suffering from mental stress at level 3. Considering only the FD phase, this was the only day that he suffered from such a

high level both of stress and of fault generation. He might have been unaware of his mental stress at stress level 3. In spite of his report, the six faults caused by human nature should have been classified into those caused by mental stress.

This conclusion seems to contradict the results that stress causes the number of faults caused by human nature to decrease, as described in 3.3 (f) and 4.1 (d). However, it can be resolved with consideration that above conclusion means that more of the faults caused by human nature already reduced by stress should be included in the total caused by stress.

4.3 Effect of physical stress on fault generation compared to mental stress

To clarify the effect of physical stress on fault generation, we counted the number of faults caused by dynamic error factors generated by developers A2 and B2 at each physical stress level (Table 10), since only these two suffered from poor physical condition in this experiment, as shown in Figure 3. By calculating the fault generation rate, we found that only developer A2 seemed to generate faults due to physical stress at physical stress level 3. He had a poor health at physical stress level 3 in the middle of the FD phase, while his mental stress (schedule pressure) values during this period were lower than or equal to his physical stress values (Figure 6). An interview with developer A2 disclosed that he had caught a cold during that period, and his poor health caused by the cold was the main factor for generating four faults during that time. Therefore, we categorized these four faults as faults caused by physical stress and excluded them from Table 4. On the other hand, developer B2 also had a poor health for a few days, but there was no correlation between physical stress levels and the number of faults during that period.

Table 10 Faults caused by dynamic error factors classified by physical stress level

Phase Condition	Developer A2				Developer B2		
	Stress Level (Physical Effort)	Accumulated (man-hours)	Generated Faults		Accumulated (man-hours)	Generated Faults	
			Caused by Stress	Caused by Human Nature		Caused by Stress	Caused by Human Nature
FD	0	15.25	2 (0.13)	6 (0.39)	27.50	0 (0.00)	1 (0.06)
	1	31.50	1 (0.03)	4 (0.13)	37.50	2 (0.09)	0 (0.00)
	2	7.75	1 (0.13)	2 (0.26)	14.00	1 (0.11)	2 (0.23)
	3	3.00	4 (1.33)	0 (0.00)	12.00	0 (0.00)	0 (0.00)
	4	--	--	--	2.00	0 (0.00)	0 (0.00)
Total		57.50	8 (0.14)	12 (0.21)	93.00	3 (0.05)	3 (0.05)
DD	0				143.25	26.5 (0.24)	13 (0.12)
	1				1.50	0 (0.00)	0 (0.00)
	2				3.75	0 (0.00)	0 (0.00)
	3				0.00	--	--
	4				9.00	1 (0.14)	2 (0.29)
Total					157.50	27.5 (0.22)	15 (0.12)

(): fault generation rate (faults/man-hour); Team B is normalized by the ratio of total effort of Team B to that of Team A

While high mental stress is considered to be 'constantly' loaded on developers throughout the software development cycle, whether accidentally or, as in the case of this experiment, intentionally, high physical stress tends to appear suddenly and accidentally for a short period. In this experiment, physical stress measured with the 'physical condition' metric appeared with two developers only for a short period. Although poor health possibly derived from both

'inner' factors such as fatigue and 'outer' factors such as catching a cold, the latter is considered to be the dominant factor, as shown in this experiment.

From these observations, we conclude as follows:

- (a) Constant mental stress is one of the most significant factors for generating faults, and
- (b) Physical stress can generate faults at a higher rate than mental stress when it appears even for a short period. It may be possible to use a physical metric for estimating faults caused by physical stress, taking into consideration the individual differences of the effect.

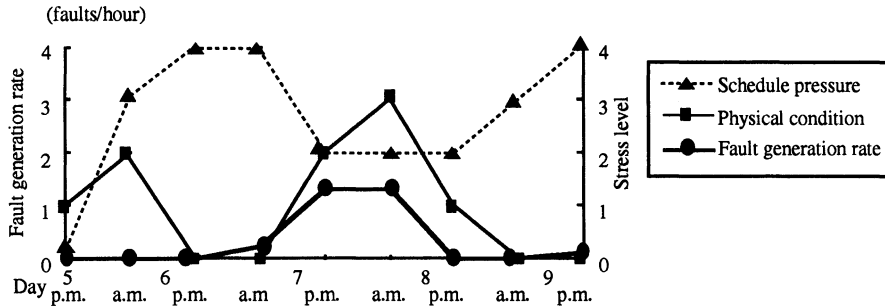


Figure 6 Physical condition and generation rate of faults caused by stress of Developer A2 in the FD phase.

5 CONCLUSIONS

As too much stress causes serious fault generation, it is important to remove it during software development to improve software reliability. The effect of stress may vary depending on factors in the development environment such as design methodology, working space, noise level around developers, and so on. To clarify the difference in stress effect depending on the development environment, we investigated the effect of mental and physical stress between different design methodologies. A controlled software development experiment was used to quantitatively determine the effect. The results are as follows.

- (a) In the functional design phase, there was no difference in fault generation caused by mental stress between Team A, which used functional design methodology, and Team B, which used structured design methodology.
- (b) In the detail design phase, however, Team A generated many more faults due to mental stress (hence, faults caused by dynamic error factors) than Team B, though Team A generated fewer faults by human nature than Team B.
- (c) Among the faults caused by human nature, many seemed to be correlated to stress at higher levels, though developers themselves were not aware of it. Therefore, faults caused by stress were considered to be generated much more often than they were actually reported by developers.
- (d) Physical stress could generate faults as well as mental stress, and it sometimes caused fault generation at a higher rate than mental stress when it appeared even for a short period.

These results lead to the conclusion that it is important not only to establish better development circumstances, but also to use more structured design methodology rather than functional design methodology to improve reliability even under sometimes inevitable stressful circumstances.

REFERENCES

- Basili, V.R. and Perricone, B.T. (1984) Software errors and complexity: an empirical investigation. *Communications of the ACM*, **27**, 42-52.
- Fujigaki, Y. (1992) Mental workload of software engineers. *Institute for Science of Labour Publishers* (in Japanese)
- Furuyama, T., Arai, Y. and Iio, K. (1994-a) Fault generation model and mental stress effect analysis. *J. Systems and Software*, **26**, 31-42.
- Furuyama, T., Arai Y. and Iio K. (1994-b) Metrics for measuring the effect of mental stress on fault generation during software development. *Int. J. Reliability, Quality and Safety Eng.*, **1**, 257-75.
- Lee, K.W., Tillman F.A. and Higgins, J.J. (1988) A literature survey of the human reliability component in a man-machine system. *IEEE Trans. Reliability*, **R-37**, 24-34.
- Nakajo, T. and Kume, H. (1991) A case history analysis of software error cause-effect relationship. *IEEE Trans. Software Eng.*, **SE-17**, 830-7.
- Oka, A., Yamamoto, S. and Isoda, S. (1993) An experimental evaluation of structured analysis and design methodology. *Trans. of Information Processing Society of Japan*, **34**, 2543-51 (in Japanese).
- Rasmussen, J., Duncan, K. and Leplat, J. (ed.) (1987) *New Technology and Human Error. John Wiley & Sons Ltd.*
- Reason, J. (1990) *Human Error. Cambridge Univ. Press.*
- Takahashi, K., Oka, A., Yamamoto, S. and Isoda, S. (1995) A comparative study of structured and text-oriented analysis and design methodologies. *J. Systems and Software*, **28**, 69-75.
- Takahashi, M and Kamayachi, Y. (1985) An empirical study of a model for program error prediction. *Proc. 8th ICSE*, 330-6.

BIOGRAPHY

Dr. Tsuneo Furuyama is a Senior Research Engineer of NTT (Nippon Telegraph and Telephone Corp.) Software Laboratories. He received the B.S. degree, the M.S. degree and the Ph. D. degree in measurement engineering from the University of Tokyo in 1968, 1970 and 1974, respectively. He has been working at NTT Electrical Communication Laboratories since 1973 and was engaged in developing programming language processors. He is now researching on software metrics. He is also a Visiting Professor of the postgraduate course of the University of Electro-communications. He is a regular member of the Information Processing Society of Japan, the Institute of Electronics, Information and Communication Engineers of Japan.

Mr. Yoshio Arai is a Senior Research Engineer of NTT software Laboratories. He received the B.S. degree in Electrical Engineering from Muroran Technical University, Japan, in 1970. He has been working at NTT Electricasl Communication Laboratories since 1970. He is now researching Software Quality Metrics. His interests include Programming Language, Software Reuse, Software Tools and Software Metrics. He is a member of the Information Processing Society of Japan.

Mr. Kazuhiko Iio is a member of NTT Software Labotratories. He received the B.S. degree in Electronics and Communication Engineering from Waseda University, Tokyo, 1989. Since 1989, he has been with Software laboratories. His current research interest whthin the Software Engineering area include the complexity of Software Metrics and Human Factors involved in Software Development. He is a member of the Information Processing Society of Japan, the Japan Society of Software Science and Technology and Japanese Society for Artificial Intelligence.