A METHODOLOGY TO PROLONG SYSTEM LIFESPAN AND ITS APPLICATION TO IT SYSTEMS

Takafumi Nakamura[1]*, Kyoich Kijima[2]

[1] Fujitsu Fsas Inc., Support Technology Group, Masonic38 MT Building, 4-1-4, Shibakoen, Minato-ku, JAPAN, nakamura.takafu@jp.fujitsu.com

[2]Tokyo Institute of Technology, Graduate School of Decision Science and Technology, 2-12-1 Ookayama, Meguro-ku, JAPAN, kijima@valdes.titech.ac.jp

*Correspondence can be directed to this author, as the primary contact.

Abstract

A system failure model to prolong system lifespan is proposed, for the purpose of preventing further occurrence of these failures. The authors claim such a methodology should have three features. First it should clarify the structure of failure factors, second it should surface hidden failure factors using statistic method especially corresponding analysis and finding the way to change. The proposed methodology is fundamentally different from the one to identify the root cause of the system failures in the sense of that it encompasses system failures as a group not as a single event. An understanding system failure correctly is crucial to preventing further occurrence of system failures. Quick fixes can even damage organizational performance to a level worse than the original state. In this sense the proposed methodology is applicable over the long time spans and therefore could be useful to confirm the effectiveness of the counter measures without introducing any side effects. Then an application example in IT engineering demonstrates that the proposed methodology proactively prolong system life learning from previous system failures.

Key words: system failure model, structuring methodology, double loop learning, ISM, risk management

1. Introduction

The purpose of this paper is to confirm the effectiveness of a proposed methodology by learning from previous system failures. The proposed methodology called Failure Factor Structuring Methodology (FFSM) is applied to PC server system failure. (Nakamura, Kijima, 2008a) In this paper we reapply FFSM to same PC server system failure after certain time period to confirm the effects of counter measure. Perrow (Perrow, 1999) argues that the conventional engineering approach to ensure safety – building in more warnings and safeguards – fails because system complexity makes failures inevitable. This indicates that we need a new model that can manage the system failure. Reason (Reason, 1997, 2004) explains the organizational life span between protection and catastrophe. The lifespan of a hypothetical organization through production-protection space (Figure 1) explains why organizational accidents repeat, with this history ending in catastrophe. This is why the periodic application of the methodology in order to prolong system life cycle.

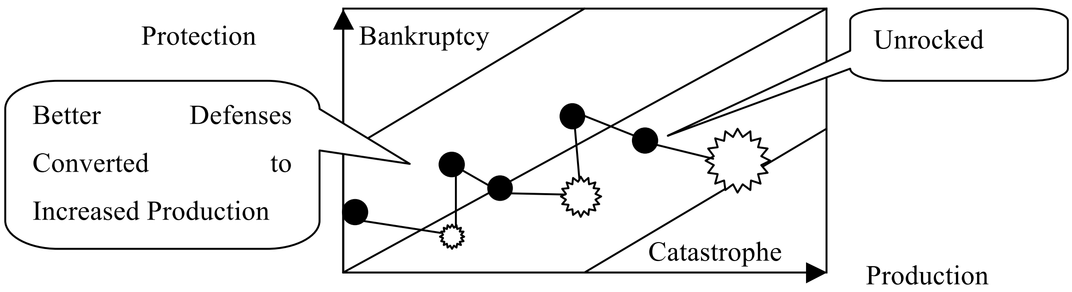

Figure 1. Lifespan of a hypothetical organization through production-protection space

Firstly we review and summarize three system failure models and clarify the features of FFSM, then discuss the results of application to PC server extended down time incidents over the two periods. Lastly we confirm that FFSM actually prolong system lifespan and its effectiveness to navigate on unrocked boat.

2. Three system failures models

In order to understand system failures, we need models and classification. Then methodologies are developed depending upon those classification. First, we introduce three system failure models with classification then introduce relating methodologies.

2.1 Simple linear system failure model (Domino model)

The archetype of a simple linear model explains system failure as the linear propagation of a chain of causes and effects (Heinrich et al., 1989). Figure 2 shows the domino metaphor for this model. The underlying principle is that system failure development is deterministic and there must have cause effect links. FTA (IEC 61025 (2006)) and FMEA (IEC 60812 (2006)) are the representative methodologies. They follow backward and forward chain respectively.

Figure 2. Domino metaphor

2.2 Complex linear system failure model (Swiss cheese model)

The archetype of a complex linear model is well known Swiss cheese model (Figure 3) first proposed by Reason (1997, 2004). The model put the importance on latent as well as manifested causes. The authors proposed FFSM (Nakamura, Kijima, 2008a) as surfacing hidden (latent) factors to suppress deviations leading to system failures.

Figure 3. Swiss cheese metaphor

2.3 Non linear or Systemic model

Rasmussen (1997) claims that systems designed according to the defense-in-depth strategy, the defenses are likely to degenerate systematically through time, when pressure toward cost-effectiveness is dominating. Correspondingly, it is often concluded by accident investigations that the particular accident was actually waiting for its release (Rasmussen, 1997). Under the presence of strong gradients behavior will very likely migrate toward the boundary of acceptable (Figure 4).

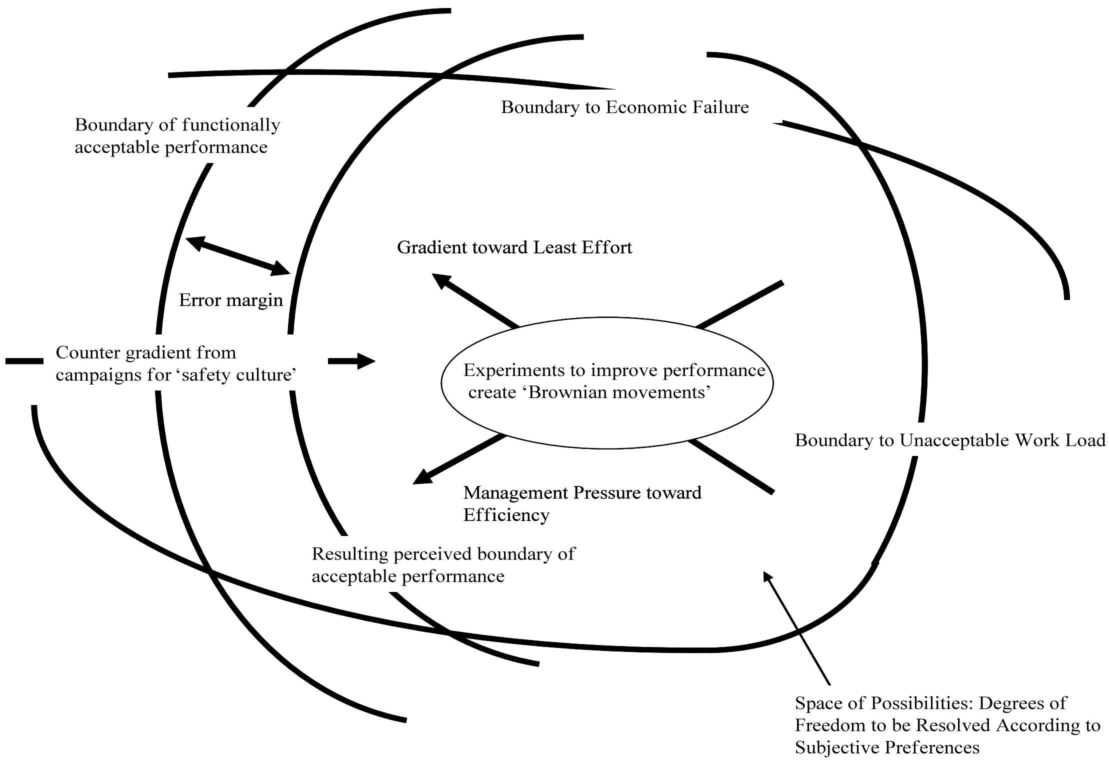

Figure 4. Rasmussen's gradient model

The authors claim that in order to indentify root causes we need to classify system failures depending upon system boundary and responsible system hierarchy introduced VSM model (Beer, 1979, 1981). The failure classes are logically identified according to the following criteria (Nakamura, Kijima, 2008b, 2009ab):

Class 1 (Failure of deviance): The root causes are within the system boundary, and conventional troubleshooting techniques are applicable and effective.

Class 2 (Failure of interface): The root causes are outside the system boundary but predictable at the design phase.

Class 3 (Failure of foresight): The root causes are outside the system boundary and unpredictable at the design phase.

Non linear systemic model is proposed as SFDM based upon system failure class (Nakamura, Kijima, 2008b, 2009a) Turner and Pidgeon found that failure responsible organization had "failure of foresight" in common. The disaster had long "incubation period" characterized by a number of discrepant events signaling potential danger. These events were typically overlooked or misinterpreted, accumulating unnoticed. In order to clarify that mechanism, Turner and Pidgeon decompose time horizon from initial stage to cultural readjustment through catastrophic disasters into six stages (Turner, Pidgeon, 1997, pp.88). Table 1 shows the feature of each stage and its relation between six stages, Failure Classes explained above. According to Six stage model, System failures have specific features corresponding to the stages. Especially failure Class 1 is located early stage of the system lifecycle, then gradually Class 2 and 3 are emerged through time. If we have methodology to monitor failure class, then we have chance to prolong system life cycle. Such a methodology can monitor system failure class and introduce counter measures. That exercise should be done periodically to see if the system stays in the right course (Reason, 1997, 2004) and without any side effects.

Table 1. Six stages of development system failures and its relation to safety archetypes

State of development	Feature	Failure Class
Stage I Initial beliefs and norms	Failure to comply with existing regulations Class1	Class1
Stage II Incubation period	Events unnoticed or misunderstood because of erroneous assumptions	Class3
		Class2 and 3
	Events unnoticed or misunderstood because of difficulties of handling information in complex situations	Class2
	Effective violation of precautions passing unnoticed because of ‘cultural –lag’ in formal precautions	Class1 and 3
	Events unnoticed or misunderstood because of a reluctance to fear the worst outcome	Class3
Stage III Precipitating event	—	—
Stage IV Onset	—	—
Stage V Rescue and salvage	—	—
Stage VI Full cultural readjustment	The establishment of a new level of precautions and expectations	Class3

2.4 Summary of three system failure models and its relating methodologies

The authors proposed meta-methodology to cover all system failures models (SOSF). (Nakamura, Kijima, 2007, 2008b, 2009ab) SOSF is derived form SOSM (Jackson, 2003) and system failure classes. SOSM classifies objects world into two dimensions. One is system and the other is participants. System dimension has two domains that are simple and complex. Participant dimension has three domains that are unitary, plural and coercive. Therefore SOSM classify the object world in to six domains (i.e. 2 x 3). And there are appropriate methodologies belonging to each domain. SOSF is complementally covers the domains based upon the worldview to see the objects system failures. Table 2 summaries above mentioned the system failure models and relating methodologies as well as meta-methodology.

Table 2. Three system failure models and its approach to management

System Failure model :Metaphor	SOSM Domain	Management Principle	Methodology	Meta-Methodology
Sequential Domino	Simple- Unitary	Eliminate Error	FTA (IEC61025), FMEA (IEC60812)	SOSF (Nakamura, Kijima, 2009ab)
Epidemiological Swiss cheese	Unitary	Find out Deviation	FFSM (Nakamura, Kijima, 2008a)	
Systemic Unrocking Boat Rasmussen's Gradients	Plural	Balancing Variability	SFDM (Nakamura, Kijima, 2008b), Six Stages (Turner, 1997)	

3. Introduction of Failure Factor Structuring Methodology

Generally, complex system failures arise from a variety of factors and combinations of those factors. And those factors have often qualitative natures. Therefore it is very important to have a holistic view by revealing quantitative relations between qualitative factors in order to construct effective methodology. The methodology should also address complex system failures in terms of obtaining the observations needed to rectify the worldview of maintenance (i.e. double loop learning). In summary such methodology should have three features. First it should clarify the structure of failure factors, second it should surface hidden failure factors using statistic method especially corresponding analysis and finding the way to change. Therefore FFSM (Nakamura, Kijima, 2008a) is the methodology to promote double loop learning through viewing the system in a holistic way. Figure 5 illustrates a general overview of such a methodology, while Table 3 clarifies the objectives of the each phase.

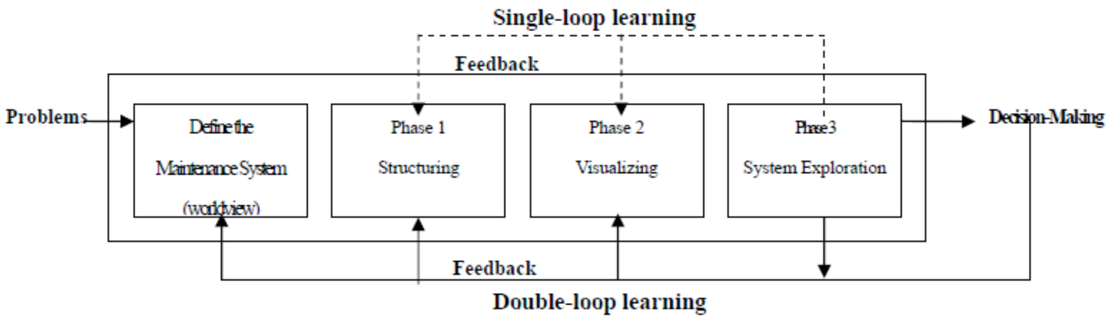

Figure 5. General overview of FFSM

Table 3. Objectives of phases 1, 2, and 3

	Feature	Objective
Phase 1	● Holistic approach (Structuring factor relationships)	Discover root causes by clarifying the relationships between factors
Phase 2	● Holistic approach (Grouping factors and problems)	Extract hidden factors behind complex symptoms by grouping factors and problems
Phase 3	● Viewing a system from a conceptual as well as a real world viewpoint ● Double-loop learning	Discover preventative measures for emergent properties by mapping factors into maintenance subsystems

4. Application to PC server failures (Extended Downtime Analysis)

This section describes the application of FFSM (Nakamura, Kijima, 2008a) to a PC server's maintenance system that manages extended downtime incidents, and explains the result of the application. It is necessary to clarify the structures and appropriate quantitative weight of each factor leading to extended downtime by analyzing PC server incidents that occurred during a given period. There are two period in this research. Period I is from April to July at 2004 and Period II from April to March at 2007 (table 4). Between Period I and II, the counter measure to foster hybrid engineer is provided based upon the outcome of FFSM applied to period I (Nakamura, Kijima, 2008a). Then evaluate the outcome of FFSM between the two periods.

Table 4. Sample incidents number

	I: April to July at 2004	II: April to March at 2007
Sample Incident Number	58	192

The overview to apply FFSM is shown in Figure 6.

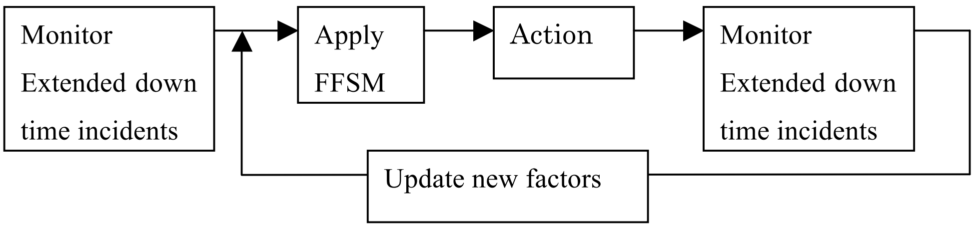

Figure 6. Application scenario of FFSM

i) Sample incidents for the period I

Number of samples: PC server extended downtime incidents (58) within the period (More than three hours from detection to resume normal operation)

The following data classification was applied to each incident to produce an Incident-Factor matrix (58 × 8). (see Appendix 1) All incidents were related to an appropriate factor(s) form eight extended downtime factors. The eight factors are defined as follows. They were extracted from the experience-based knowledge of engineers with previous experience of extended downtime incidents.

S_1: Product

S_2: Isolation (Diagnose faulty parts)

S_3: Maintenance Organization (Skills, Scale and Deployment)

S_4: Spare Parts (Deployment and Logistics)

S_5: Faulty Spare Parts

S_6: Fix has not applied (EC has not applied)

S_7: Recovery Process

S_8: Software bug

ii) Sample incidents for the period II

Number of samples: PC server extended downtime incidents (192) within the period (More than three hours from detection to resume normal operation)

The following data classification was applied to each incident to produce an Incident-Factor matrix (192 × 9). (see Appendix 2) S_9: Human Error (Operation etc) is added based upon the outcome of FFSM applied to period I (Nakamura, Kijima, 2008a).

4.1 Phase 1 transition of failure factor structure

This phase enables structuring of causes of a system failure. Then it is required to reveal quantitative factor relations from qualitative factors. To achieve the feature, we have applied ISM

(Sage, 1977; Warfield, 1976, 1980) for this phase. Figure 7 and Figure 8 shows the direct influential matrix X* that is obtained by analyzing the causal relationship between the eight and nine factors (from S_1 to S_9). The direct influential matrix is the causal relation's matrix in which the columns and rows contain the factors from S_1 to S_9.

$X^* = (x_{jk})$: x_{jk}= 3, 2, 1 (if there is a direct causal relationship form column j to row k)

[3: strong relationship, 2: moderate relationship, 1: weak relationship]

x_{jk}= space (if there is no direct causal relationship form column j to row k)

In period II (Figure 8) the two factors S_3 and S_6 are disappeared due to the action taken between the two periods. And new factor S_9 is newly introduced. Therefore the direct influential matrix size is diminished from 8 x 8 to 7 x 7.

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
S_1		3			2	1	3	2
S_2							3	
S_3		2				2	3	
S_4					1		1	
S_5		2		1			1	1
S_6		1			2		1	
S_7								
S_8		3			1		3	

Figure 7. Direct influential matrix X* (Period I)

	S_1	S_2	S_4	S_5	S_7	S_8	S_9
S_1		3		2	3	2	3
S_2					3		2
S_4				1	1		
S_5		2	1		1	1	1
S_7							3
S_8		3		1	3		3
S_9							

Figure 8. Direct influential matrix X* (Period II)

The Figure 13 and Figure 14 show the difference of the structures between the factors. The upper level is the root cause to the lower levels. Therefore the countermeasures to the upper level are more essential to that of lower levels. S_3: Maintenance Organization (Skills, Scale and Deployment) and S_6: Fix has not applied (EC has not applied) are eliminated and S_9: Human Error (Operation etc) is added in the period II. The action taken between the two periods caused the transition of failure factor structure. The upper most factor of S_3 does no longer exist and the lowest factor of S_9 is appeared. The number attached on the arrow is calculated for Z as indirect influence Matrix which influences all the indirect relation between the factors. To consider indirect causes in causal analysis, it is necessary to introduce the normalized direct influential matrix X. Figure 9 is the normalized direct influential matrix X, which is obtained by dividing the maximum load factor (11) that is obtained from the maximum value within the summation of each column of X* (Figure 7). Then, the total influential matrix Z (see Figure 10) that includes the indirect cause can be obtained based upon the following operation of X.

$$Z=X+X^2+X^3+\ldots=X*(I-X)^{-1}$$

The element of Z represents the relative weight of each causal relation.

Figure 13 shows the overall structure of the eight factors in five levels. The number attached to the each arrow represents the element of Z (see Figure 10). The same processes are applied to the period II and the results are shown in Figure 8, Figure 11, Figure 12 and Figure 14.

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
S_1		0.27			0.18	0.09	0.27	0.18
S_2							0.27	
S_3		0.18				0.18	0.27	
S_4					0.09		0.09	
S_5		0.18		0.09			0.09	0.09
S_6		0.09			0.18		0.09	
S_7								
S_8		0.27			0.09		0.27	

Figure 9. Normalized direct influential matrix X (Period I)

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
S_1		0.37		0.02	0.21	0.09	0.45	0.20
S_2							0.27	
S_3		0.21			0.03	0.18	0.35	
S_4		0.02		0.01	0.09		0.11	0.01
S_5		0.20		0.09	0.02		0.18	0.09
S_6		0.12		0.02	0.18		0.14	0.02
S_7								
S_8		0.29		0.01	0.09		0.36	0.01

Figure 10. Total influential matrix Z (Period I)

	S_1	S_2	S_4	S_5	S_7	S_8	S_9
S_1		0.27		0.15	0.23	0.15	0.23
S_2					0.23		0.15
S_4				0.08	0.08		
S_5		0.15	0.08		0.08	0.08	0.08
S_7							0.23
S_8		0.23		0.08	0.23		0.23
S_9							

Figure 11. Normalized direct influential matrix X (Period II)

	S_1	S_2	S_4	S_5	S_7	S_8	S_9
S_1		0.29	0.01	0.16	0.35	0.16	0.40
S_2					0.23		0.20
S_4		0.01	0.01	0.08	0.09	0.01	0.03
S_5		0.17	0.08	0.01	0.15	0.08	0.16
S_7							0.23
S_8		0.24	0.01	0.08	0.29	0.01	0.34
S_9			.				

Figure 12. Total influential matrix Z (Period II)

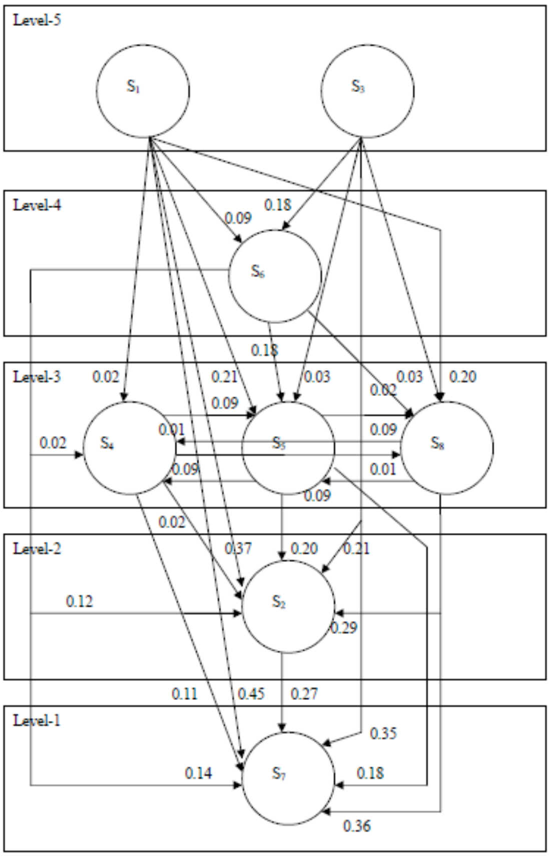

Figure 13. Overall structure of eight factors

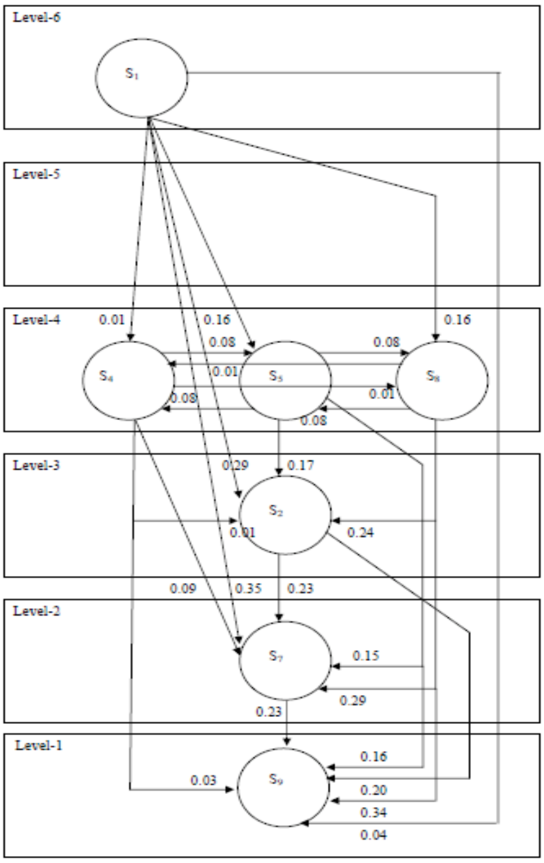

Figure 14. Overall structure of seven factors

4.2 Phase 2 transition of hidden failure factors

This phase enables to find hidden factors that are not extracted by analyzing each specific failure event. Our idea is to adopt the quantification theory type III (Hayashi, 1952; Gifi, 1990; Van de Geer, 1993; Greenacre, 1984, 1983) to find such hidden factors. This method is one of the correspondence analyses (Greenacre, 1984, 1993) and useful to quantify and visualize entire failure factors that have qualitative nature. A PC program named 'excel toukei 2002' (Kabushiki-kaishiya Shiyakai-Jiyouhou-Service, 2002) was used for this analysis. The three hidden factors extracting by phase 2 analysis in period I are as follows.

- 1st axis Isolation for faulty parts
- 2nd axis Software recovery
- 3rd axis Hardware maintenance organization

Table 5. Factor axes and attributes (Period I)

	Eigen value	Contribution ratio	Accumulated distribution ratio	Correlation coefficient
1st axis	0.80	19.65%	19.65%	0.89
2nd axis	0.71	17.58%	37.23%	0.84
3rd axis	0.66	16.19%	53.41%	0.81
4th axis	0.61	15.03%	68.44%	0.78
5th axis	0.53	13.19%	81.63%	0.73
6th axis	0.42	10.26%	91.89%	0.64

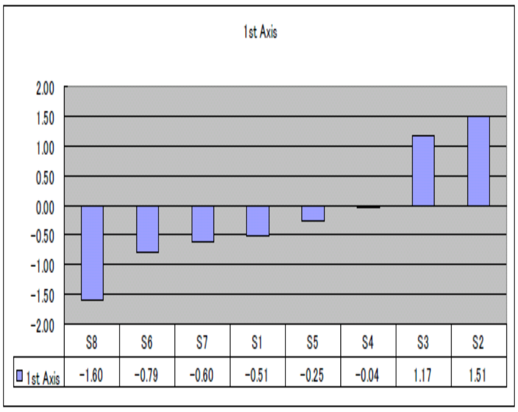

Figure 15. Factor scores for the first axis (Isolation of faulty parts)

Accumulated contribution ratios up to three axes are 53%. (Table 5) The three hidden factors cover more than 50% incidents.

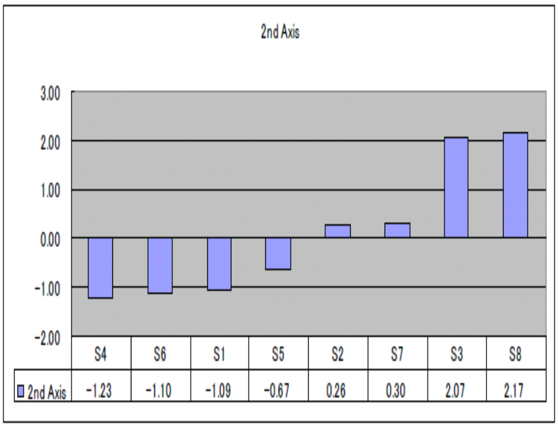

Figure 16. Factor scores for the second axis (Software recovery)

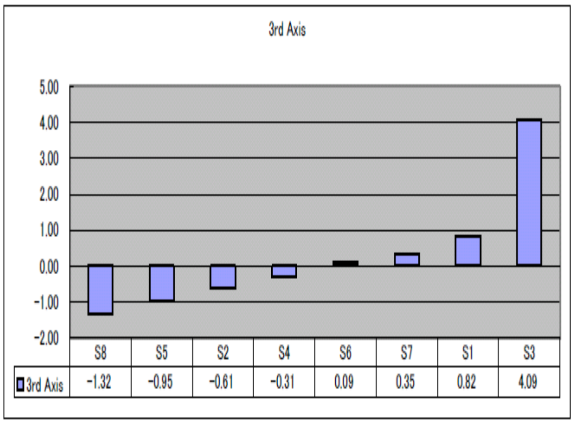

Figure 17. Factor scores for the third axis (Hardware maintenance organization)

S_2 and S_3 are located adjacently, this indicate Isolation activity and Maintenance organization have strong correlation. Therefore 1st axis is named as Isolation for faulty parts (Figure 15).

S_8 and S_3 are located adjacently, this indicate Software bug and Maintenance organization have strong correlation. Therefore 2nd axis is named as Software recovery (Figure 16).

S_3 and S_1 are located adjacently, this indicate Maintenance organization and Product have strong correlation. Therefore 3rd axis is named as Hardware maintenance organization (Figure 17).

The three hidden factors extracting by phase 2 analysis in period II are as follows.

- 1st axis Product
- 2nd axis Software recovery
- 3rd axis Human error

Table 6. Factor axes and attributes (Period II)

	Eigen value	Contribution ratio	Accumulated distribution ratio	Correlation coefficient
1st axis	0.87	19.98%	19.98%	0.93
2nd axis	0.83	19.01%	38.99%	0.91
3rd axis	0.77	17.77%	56.76%	0.88
4th axis	0.74	17.05%	73.82%	0.86
5th axis	0.69	15.82%	89.63%	0.83
6th axis	0.45	10.37%	100.00%	0.67

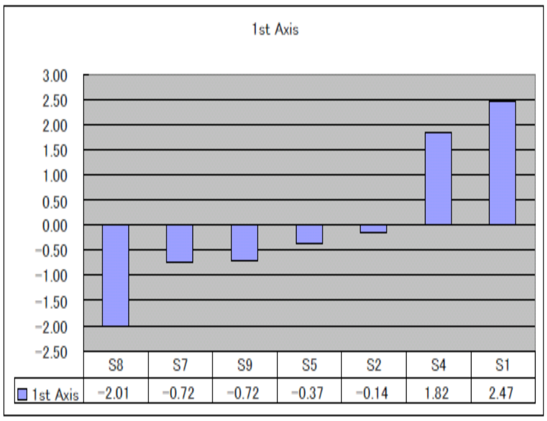

Figure 18. Factor scores for the first axis (Product)

Accumulated contribution ratios up to three axes are 57%. (Table 6) The three hidden factors cover more than 50% incidents.

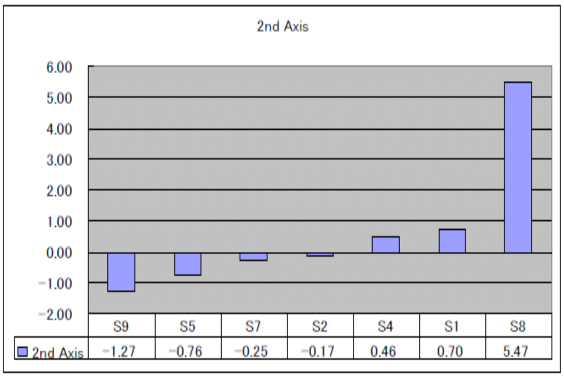

Figure 19. Factor scores for the second axis (Software recovery)

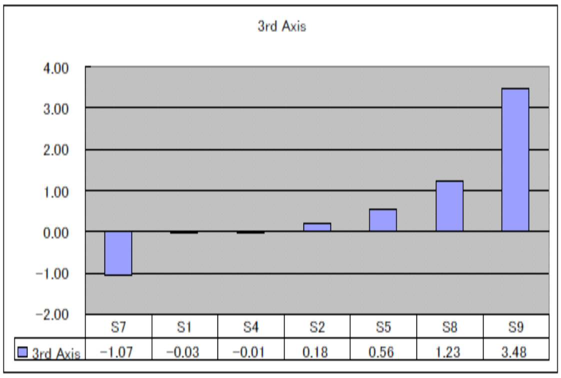

Figure 20. Factor scores for the third axis (Human error)

S_1 and S_4 are located adjacently, this indicate Product and Spare parts have strong correlation. Therefore 1st axis is named as Product (Figure 18).

S_8 and S_1 are located adjacently, this indicate Software bug and Product have strong correlation. Therefore 2nd axis is named as Software recovery (Figure 19). S_9 and S_5 are located adjacently; this is the same as 1st axis. This indicate Human error (S_9) is relating to faulty spare parts (S_5).

S_9 and S_8 are located adjacently, this indicate Human error and Software Bug have strong correlation. Therefore 1st axis is named as Human error (Figure 20).

The counter measure to foster hybrid engineer between the two periods caused two transitions. One is that the 1st axis (i.e. Isolation for faulty parts) is replaced by Product and the other is that the 3rd axis Hardware maintenance organization to Human error. The 2nd axis remains the same. The transition suggests us two points. One is to improve Product quality is crucial rather than to improve isolation faulty parts technique and the other is to reduce human error. The followings are the results of the action taken between the two periods. One is the hybrid engineer removed Organization factor and the other is the transition newly introduced human error factor.

4.3 Phase 3 the way forward

Figure 21 and Figure 22 are the outcome of phase 3 analysis in period I and II respectively. The number shown in the circle indicate responsible domain of IT system life span. There are three phases; Design, Configuration and Operation. The ideal situation is the number of incidents classified in the operation quadrant is zero. Namely all of the extended downtime incidents are suppressed in production (i.e. operation) phase. In this regards, Period I has 48 incidents (59%) have been treated class 3 failure. In this case fostering hybrid engineer is decided and applied in the real situation. Figure 22 is the result of allocation of fostering hybrid engineers. The result shows the reduction of class 3 failures form 59% to 44% due to the introduction of hybrid engineers. This causes a new class 3 failure which is human error. The counter measure should be three areas. They are i) the product enhancement to strengthen RAS (Reliability, Availability and Serviceability) as well as ii) to implement fail-safe features and iii) to educate engineers especially hybrid technology area. Table 7 shows the new learning from phase 3 analysis and Table 8 summarizes FFSM application results for Period I and II.

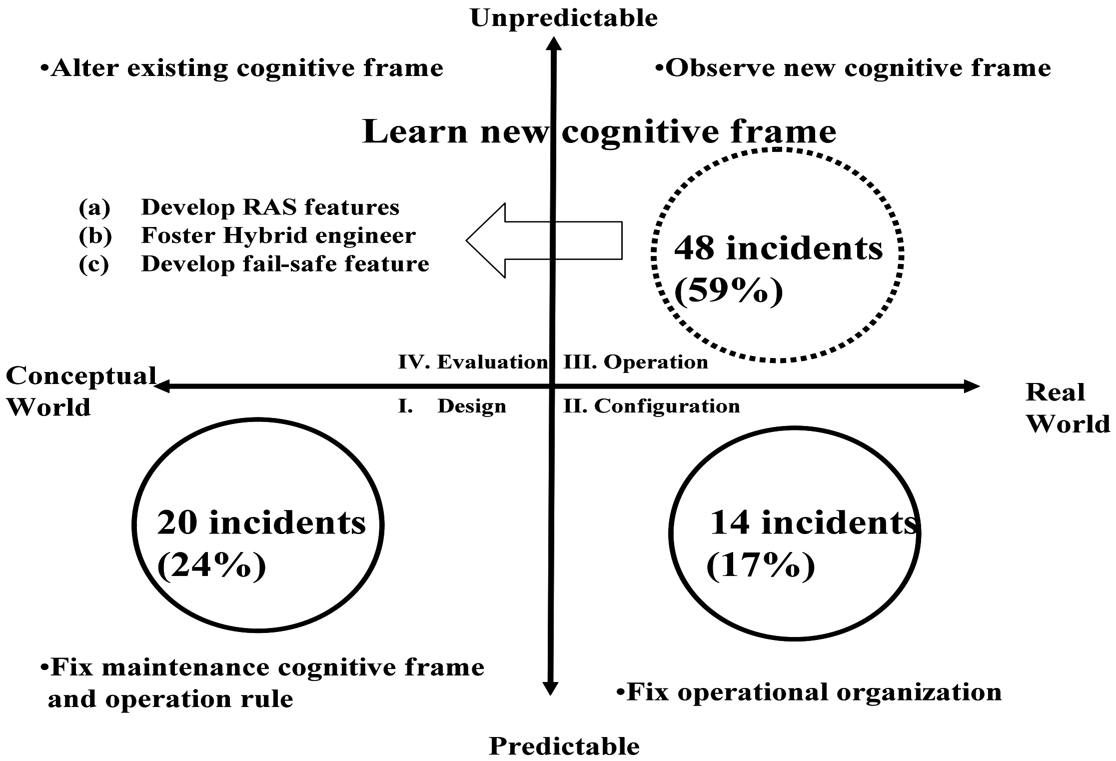

Figure 21. Factors contributing to PC server extended downtimes and a maintenance cognitive frame (Period I)

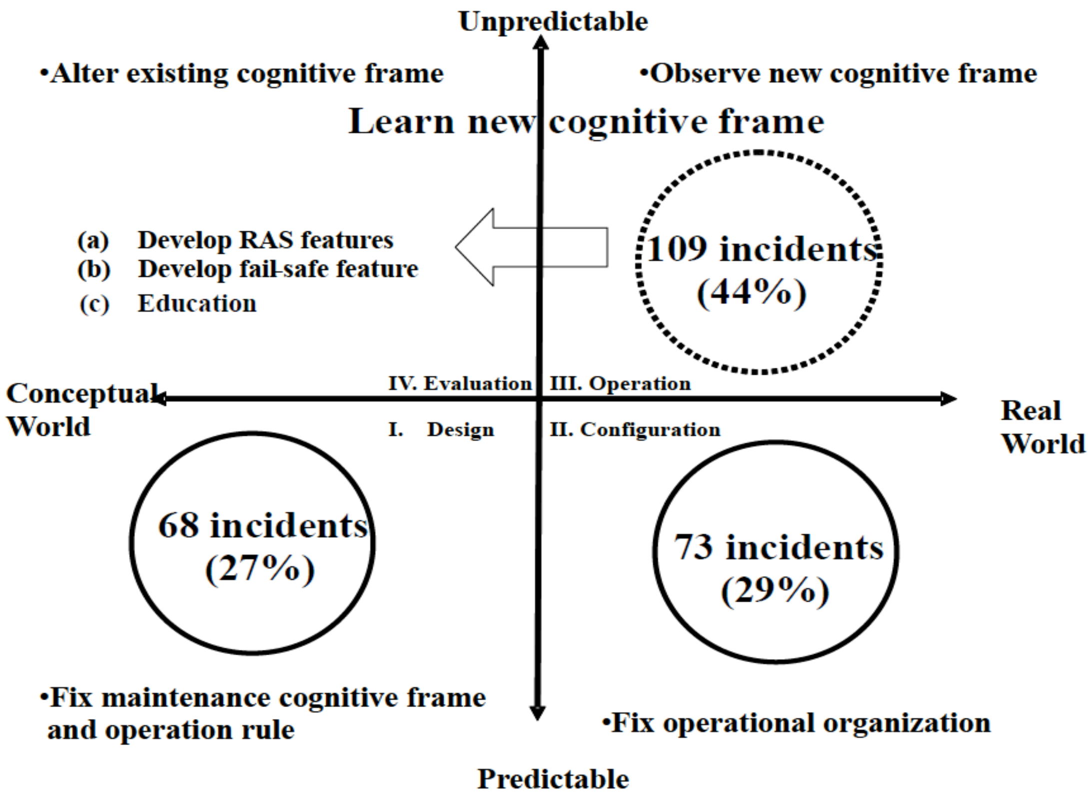

Figure 22. Factors contributing to PC server extended downtimes and a maintenance cognitive frame (Period II)

Table 7. A new cognitive frame obtained through Phase 3 analysis

	Period I	Period II
Design	24% (20 Incidents)	27% (68 Incidents)
Configuration	17% (14 Incidents)	29% (73 Incidents)
Operation	59% (48 Incidents)	44% (109 Incidents)
Evaluation	(a)Develop RAS features (b)Foster Hybrid Engineer (c)Develop fail-safe feature	(a)Develop RAS features (b)Develop fail-safe feature (c)Education
Extended down time occurrence rate	0.48% (Incidents/100shipments.year)	0.21% (Incidents/100shipments.year)

Table 8. Summary of application results of FFSM

	Objective	**Result (Period I)**	**Result (Period II)**
Phase 1	To discover root causes	● S_1 (Product) and S_3 (Maintenance organization) are the uppermost factors (i.e. root causes) and S_2 (Isolation) and S_7 (Recovery process) are the lowest factors contributing to extended downtimes ● The major upper factors (i.e. root causes) of S_7 (Recovery process) are S_1 (Product) (0.45), S_8 (Software bug) (0.36), and S_3 (Maintenance organization) (0.35). The number in parentheses indicates the relative weight of the related factor. This indicates that product and software-related maintenance organizations are the root cause for an extended period being needed for recovery. ● The major upper factors (i.e. root causes) of S_2 (Isolation) are	● S_1 (Product) is the uppermost factors (i.e. root causes) and S_9 (Human error) is the lowest factor contributing to extended downtimes ● The major upper factors (i.e. root causes) of S_9 (Human error) are S_8 (Software bug) (0.34), S_7 (Recovery process) (0.23), and S_2 (Isolation) (0.20). The number in parentheses indicates the relative weight of the related factor. This indicates that Software related recovery and problem isolation is the root cause for an extended period being needed for recovery. ● The major upper factors (i.e. root causes) of S_7 (Recovery process) are S_1 (Product) (0.35) followed by S_8

		S_1 (Product) (0.37) followed by S_8 (software bug) (0.29). The numbers in parentheses also indicate that the product and software bugs are the root causes for an extended period being needed for recovery. ● Among the sample incidents, 45% had multiple factors (i.e. 26 out of 58 incidents) (Appendix 1)	(software bug) (0.29). The numbers in parentheses also indicate that the product and software bugs are the root causes for an extended period being needed for recovery. ● Among the sample incidents, 20% had multiple factors (i.e. 38 out of 192 incidents) (Appendix 2)
Phase 2	To extract hidden factors behind complex symptoms	● The hidden factors contributing to extended downtimes have three causes represented by three axes: the 1st axis (Isolation for faulty parts), the 2nd axis (Software recovery), and the 3rd axis (Hardware maintenance organization).	● The hidden factors contributing to extended downtimes have three causes represented by three axes: the 1st axis (Product), the 2nd axis (Software recovery), and the 3rd axis (Human error).
Phase 3	To discover preventative measures for emergent properties	● Phase 3 analysis creates a new worldview of PC system maintenance (Table 7 and Figure 21) ● All three worldviews are counter-measures for emergent problems, none of which can be managed proactively in the design or configuration phase. ● The three new worldviews (Table 7) are new inputs to FFSM to confirm further improvement.	● Phase 3 analysis creates a new worldview of PC system maintenance (Table 7 and Figure 22) ● All three worldviews are counter-measures for emergent problems, none of which can be managed proactively in the design or configuration phase. ● The three new worldviews (Table 7) are new inputs to FFSM to confirm further improvement.

5. Conclusion

The application results shown in the previous chapter suggests that the next challenge is to introduce RAS function and safe guards for product itself. S_9 (Human error) is the lowest level cause of the extended down time and S_1 (Product) is the upper most factors (Figure 14). This suggest that the counter measures to the product is not the only action but to educate engineers especially hybrid technology area. And the ratio of class 3 failures has been reduced from 59% to 44% during the period. One of the counter measures of the post period I was to foster hybrid engineer. Fostering hybrid engineer is to understand the extended down time incidents as class 3 failures (i.e. to exercise the software centric activities was used to be out of the system boundary of the hardware engineers). The application results over the two periods clearly show the reduction of the class 3 failures. Also the reduction of occurrence ratio of extended down time incidents actually confirmed the effectiveness of the FFSM in IT arena. This confirmed that the progress of Turner's stages up to stage II and revert back to stage I by introducing organizational change. In this regards FFSM is one of the methodologies to prolong system lifespan as shown Unrocking boat metaphor (Reason, 2007, 2004). These outcomes are not obtained from conventional methodology. (i.e. Domino model) Table 9 shows the sustainability spectrum of learning failure factors. Right hand side of the spectrum is the key to prolong system life span. In the end the summary of the main claim point in this paper is that FFSM actually prolong the system life cycle for IT systems. It is essential to apply and reapply during the certain time periods over the transformation initiated by the counter measures. (i.e. in this paper it is to foster hybrid engineer) and also to update the failure factors (i.e. in this paper S_3: Maintenance Organization (Skills, Scale and Deployment) and S_6: Fix has not applied (EC has not applied) are eliminated and S_9: Human Error (Operation etc) is added in the period II).

Table 9. Sustainability spectrum of learning failure factors

System	Systematic	⟺	Systemic
	Static	⟺	Dynamic
Learning	Single loop	⟺	Double loop
VSM	System 1	⟺	System 5
Failure Class	Class 1	⟺	Class 3
FFSM	Phase 1	⟺	Phase 3

Appendix 1. Sample incident matrix in period I

Shading indicates that the sample has multiple factors (26 incidents)

Sample #	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
1	0	0	0	0	0	1	0	0
2	0	0	0	1	1	0	0	0
3	0	0	0	1	1	0	0	0
4	0	1	0	0	0	0	1	0
5	0	1	0	0	1	0	0	0
6	0	1	0	0	0	0	1	0
7	1	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0
10	0	0	0	1	0	0	0	0
11	1	0	0	1	0	0	0	0
12	0	0	1	0	0	0	0	0
13	0	0	0	0	0	0	1	0
14	0	0	0	0	0	0	1	0
15	0	1	0	0	0	0	0	0
16	0	0	0	0	0	0	1	0
17	0	1	0	1	0	0	0	0
18	0	1	0	0	0	0	0	0
19	0	0	0	0	0	0	1	0
20	0	0	0	0	0	0	1	1
21	0	1	1	0	0	0	0	0
22	1	0	0	0	0	0	1	0
23	0	0	0	0	0	0	1	0
24	1	0	0	0	0	0	1	0
25	1	0	0	0	0	0	1	0
26	1	0	0	0	0	0	1	0
27	0	1	0	0	0	0	0	0
28	0	1	0	0	0	0	0	0
29	0	1	0	0	0	0	0	0
30	0	1	0	0	0	0	0	0
31	0	0	0	0	0	0	1	0
32	0	0	0	0	0	1	1	0
33	0	1	0	0	0	0	0	0
34	1	1	0	0	0	0	0	0
35	0	1	0	0	0	0	1	0
36	0	1	0	0	0	0	0	0
37	0	0	0	0	1	0	1	1
38	0	0	0	0	0	0	0	1
39	0	0	0	0	0	0	0	1
40	0	1	0	1	0	0	0	0
41	0	0	0	1	0	0	1	0
42	0	0	1	0	0	0	1	0
43	1	0	0	1	0	0	1	0
44	0	0	0	0	1	1	0	0
45	0	1	0	0	0	0	0	0
46	0	1	0	0	0	0	0	0
47	1	0	0	0	0	0	0	0
48	0	0	0	0	0	0	1	1
49	0	0	0	0	0	0	0	1
50	0	0	0	0	0	0	1	0
51	1	0	0	1	0	1	0	0
52	0	1	0	0	0	0	0	0
53	0	1	0	0	0	0	0	0
54	0	0	0	1	0	0	0	0
55	0	1	0	0	0	0	0	0
56	0	0	0	0	0	0	1	1
57	0	0	0	0	0	1	1	0
58	0	1	0	0	0	0	0	0

Appendix 2. Sample incident matrix in period II

Shading indicates that the sample has multiple factors (38 incidents)

Samples # 43 to 178 are intentionally eliminated

Sample#	S1	S2	S3	S4	S5	S6	S7	S8	S9
1	0	1	0	0	0	0	1	0	0
2	0	0	0	0	0	0	0	1	0
3	0	1	0	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0
5	0	1	0	0	0	0	0	0	0
6	0	1	0	0	0	0	1	0	0
7	0	0	0	0	0	0	0	0	1
8	0	0	0	1	0	0	1	0	0
9	0	1	0	0	0	0	0	0	0
10	0	0	0	1	0	0	0	0	0
11	0	0	0	0	0	0	1	0	1
12	0	0	0	1	0	0	0	0	0
13	0	1	0	0	0	0	0	1	0
14	0	1	0	0	0	0	0	0	1
15	0	1	0	0	0	0	0	0	0
16	0	1	0	1	0	0	0	0	0
17	0	0	0	0	0	0	0	0	1
18	0	1	0	1	0	0	0	0	0
19	0	0	0	0	1	0	0	0	1
20	0	1	0	0	0	0	0	0	0
21	0	0	0	0	0	0	1	0	0
22	0	0	0	0	0	0	1	0	0
23	0	0	0	0	0	0	1	0	0
24	0	1	0	0	0	0	1	0	0
25	0	0	0	1	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0
27	0	0	0	1	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0
29	0	1	0	0	0	0	1	0	0
30	0	0	0	0	0	0	1	1	0
31	0	1	0	0	0	0	1	0	0
32	0	1	0	0	0	0	0	0	0
33	0	0	0	1	0	0	0	0	0
34	0	0	0	0	0	0	0	1	0
35	0	1	0	0	0	0	0	0	0
36	0	1	0	1	0	0	0	0	0
37	0	1	0	0	0	0	0	0	0
38	0	1	0	0	0	0	0	0	0
39	0	1	0	0	0	0	1	0	0
40	0	1	0	0	0	0	0	0	0
41	0	0	0	0	0	0	1	0	0
42	0	1	0	0	0	0	0	0	0
179	0	0	0	0	0	0	1	0	0
180	0	0	0	0	0	0	1	0	0
181	0	0	0	0	0	0	1	0	0
182	0	0	0	1	0	0	0	0	0
183	0	1	0	0	0	0	0	0	0
184	0	0	0	0	0	0	0	0	0
185	0	0	0	0	0	0	1	0	0
186	0	0	0	0	0	0	1	0	0
187	0	1	0	0	0	0	0	0	0
188	0	1	0	0	0	0	1	0	0
189	0	1	0	0	0	0	0	0	0
190	0	0	0	0	1	0	0	0	0
191	0	1	0	0	0	0	1	0	0
192	0	0	0	0	0	0	1	0	0

References

Beer, S. 1979. The Heart of Enterprise. John Wiley & Sons: London and New York.

Beer, S. 1981. Brain of the Firm, 2nd edition. John Wiley & Sons: London and New York.

Gifi, A (1990). Nonlinear multivariate analysis. Wiley

Greenacre, M. J. (1984). Theory and applications of correspondence analysis. Academic Press

Greenacre, M. J. (1993). Correspondence analysis in practice. Academic Press.

Hayashi, C (1952). Prediction of Phenomena from Qualitative Data and Quantification of Qualitative Data. Annals of the Institute of Statistical Mathematics, Vol. III.

Heinrich H.W., Petersen D., and Roos N. 1989. Industrial Accident Prevention: A Safety Management Approach. 5th ed. McGraw-Hill: New York

IEC 60812 (2006). Procedure for failure mode and effect analysis (FMEA)

IEC 61025 (2006). Fault tree analysis (FTA)

Jackson, M, C. (2003). Systems Thinking: Creative Holism for Managers. John Wiley & Sons Inc

Kabushiki-kaishiya Shiyakai-Jiyouhou-Service (2002) 'Excel Toukei 2002 for Windows'

Nakamura, T. and Kijima, K. (2007). Meta system methodology to prevent system failures. Proceeding of the 51st Annual Meeting of ISSS in Tokyo (Aug. 2007).

Nakamura, T. and Kijima, K. (2008a). A Methodology for Learning from System Failures and its Application to PC Server Maintenance. Risk Management 10.1, 2008: 1-31.

Nakamura, T. and Kijima, K. (2008b). Failure of Foresight: Learning from System Failures through Dynamic Model. Proceeding of the 52nd Annual Meeting of ISSS in Madison (Jul. 2008).

Nakamura, T. and Kijima, K. (2009a). System of system failures: Meta methodology for IT engineering safety. Systems Research and Behavioral Science Vol.26, Issue 1, January/February 2009: 29-47.

Nakamura, T. (2009b). Systems of System Failures: Double Loop Learning Meta Methodology for IT Systems, PhD dissertation, Tokyo Institute of Technology.

Perrow, C. 1999. Normal Accidents Living with High-Risk Technologies. Princeton Paperbacks: New York.

Rasmussen, J. 1997. Risk Management in a dynamic society: a modeling problem. Safety Science vol. 27, no 2/3:183-213.

Reason, J. (1997). Managing the Risk of Organizational Accident. Ashgate Pub Ltd

Reason, J. and Hobbs, A (2004). Managing maintenance Error: A practical Guide. Ashgate Pub Ltd

Sage, A, P. (1977).System Engineering-Methodology and Applications. IEEE press

Turner B.A. , Pidgeon N.F.1997. Man-Made Disasters 2nd edition. Butterworth-Heinemann. UK

Van de Geer, J. P (1993). Multivariate analysis of categorical data: Application. Sage.

Warfield, J, N. (1980, June). ISM and Related Work-annotated bibliography, Dept.of E.E., Univ.of Virginia

Warfield, J, N. (1976). Societal Systems PLANNING, POLICY AND COMPLEXITY, John Wiley & Sons