FAILURE OF FORESIGHT: LEARNING FROM SYSTEM FAILURES THROUGH DYNAMIC MODEL

Takafumi Nakamura[1]*, Kyoich Kijima[2]
[1]Tokyo Institute of Technology, Graduate School of Decision Science and Technology,
2-12-1 Ookayama, Meguro-ku, JAPAN, nakamura.takafu@jp.fujitsu.com
[1] Fujitsu Fsas Inc., Support Technology Group, Masonic38 MT Building, 4-1-4, Shibakoen, Minato-ku, JAPAN
[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 dynamic model for holistically examining system failures is proposed, for the purpose of preventing further occurrence of these failures. 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. There is well known side effect of “normalized deviance” which leads NASA’s Challenger and Columbia space shuttle disasters. And there is so called “incubation period” which leads to catastrophic system failures in the end. However this indicates there is a good chance to avoid catastrophic system failures if we can sense the incubation period correctly and respond the normalized deviance effect properly. If we don’t understand system failure correctly, we can’t solve it effectively. Therefore we first define three failure classes to treat dynamic aspects of system failures. They are Class 1 (Failure of deviance), Class 2 (Failure of interface) and Class 3 (Failure of foresight) respectively. Then we propose a dynamic model to understand system failure dynamically through turning hindsight to foresight to prevent further occurrence. An application example in IT engineering demonstrates that the proposed model proactively promotes double loop learning from previous system failures.

Key words: system failure, engineering safety, dynamic model, double loop learning

1. INTRODUCTION

The purpose of this paper is to propose a dynamic model to promote engineering safety by learning from previous system failures. The predominant worldview in IT engineering is that systems failures can be prevented at the design phase. This worldview is obvious if we examine mainstream, current methodologies for managing system failures. These methodologies use a reductionist approach and are based on a static model (Nakamura, Kijima, 2008). It is often pointed out that most such methodologies have difficulty coping with emergent properties in a proactive manner and preventing the introduction of various side effects from quick (i.e., temporary) fixes, which leads to repeating failures of similar type. The main reason for this situation is that current methodologies tend to identify a system failure as a single, static event, so organizational learning tends to be limited to a single loop rather than a double loop in rectifying the model of the model (i.e., the meta model) of action (i.e., the operating norm). Double loop learning skill should enable people to question basic assumptions, which leads to modifying mental models to create action producing desired goals, rather than simply modifying actions under current mental models (Morgan, 1986; Argyris, Schoen,1996; Senge, 1990).

Heinrich's law (Heinrich, Petersen, and Roos, 1989) which is well known in the industrial world that state there are 29 minor injuries and 300 troubles in the background of a serious injury. This indicates that there are enough signs prior to a severe system failure or a serious injury. However explanations of system failures in terms of a reductionist approach (i.e., an event chain of actions and errors) are not very useful for designing improved systems (Rasmussen, 1997; Leveson, 2004). In addition, 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 dynamic aspects of system failure, by ensuring the efficacy of its countermeasures through the promotion of double loop learning.

In this paper, we propose a new way to interpret system failures dynamically in order to overcome the current methodologies' shortcomings. We also demonstrate the proposed model's efficacy through an application in IT engineering.

2. TAXONOMY OF SYSTEM FAILURES AND INTRODUCTION OF THREE FAILURE CLASSES

Prior to explain safety archetypes we need to review taxonomy of system failures. We should have a common language for understanding system failure objectively. It is vital to examine system failure from various perspectives. System safety can be achieved through the actions of various stakeholders. One such common language was developed by van Gigch (van Gigch, 1986) for taxonomy of system failures. There are six categories of system failures. They are failure of i) technology, ii) behavior, iii) structure and regulation, iv) rationality and v) evolution. However common language is not adequate to treat dynamic aspect of system failures. Furthermore we need to introduce three failure classes in order to avoid the dynamic aspects of system failures (i.e., erosion of safety goals over time). The failure classes should intentionally be identified in conjunction with the VSM model (Beer,1979,1981). They should clarify the system boundary and the nature of a problem (i.e., predictable or unpredictable). The failure classes are logically identified according to the following criteria:

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.

The failure classes thus depend on whether the root causes are inside or outside the system boundary, and a class-3 failure for one person can be a class-1 or -2 failure for other people. Therefore, the definition is relative and recursive, so it is important to identify the problem owner, in terms of two aspects: the stakeholder group, and the VSM system (i.e., systems 1 to 5). Unless those two aspects are clarified, failure classes cannot be identified.

It is necessary to recognize the organizational system level in order to rectify the operational norm, because for preventing further occurrence of system failures, it is inadequate to change only systems 1 to 3 (or the phase system for seeking when and how). As pointed out above, current technological models mainly focus on the operational area, and this can lead to side effects of quick fixes. Event chain models developed to explain system failures usually concentrate on the proximate events immediately preceding the failures. The foundation of a system failure, however, is often laid years before the failure occurs. In this situation, the VSM model serves well for understanding real root causes.

In a stable environment, control of activities and their safety by a prescriptive manual approach deriving rules of conduct from the top down can be effective. In the present dynamic environment, however, this static approach is inadequate, and a fundamentally different view of system modeling is required. Next Section describes dynamic model (i.e. Safety Archetypes) explaining why fixing failures sometimes introduces unintended side effects and how dynamic understanding contributes to introducing ultimate counter measures.

3. UNDERSTANDING SYSTEM FAILURE THROUGH DYNAMIC MODEL

The frequent occurrence of deviant system failures has become regular but poorly understood. For example, deviant system failure is believed to lead to NASA's Challenger and Columbia space shuttle disasters (Columbia Accident Investigation Board Report, Chapter 6, pp. 130). This normalized deviance effect is hard to understand from a static failure analysis model. NASA points out the notion of "History as Cause" for repeated disastrous failures (Columbia Accident Investigation Board Report, Chapter 8). And this normalized deviance is tightly relating so called "incubation period" prior to catastrophic disasters (Turner, Pidgeon, 1997; Vaughan, 1997).

These considerations imply usefulness to focus on the dynamic aspects of the cause and effect of system failures rather than the static aspects. Dynamic model analysis is applicable in all technology arenas, including high-risk technology domains like that of NASA. There are some pitfalls, however, in introducing countermeasures. Quick fixes seem to work in a short time span but gradually have a saturated effect in the long term or can even damage organizational performance to a level worse than the original state. This can be explained using a dynamic model of the safety archetype.

There are well-known archetypes of fixes that fail, eroding safety goals and degrading the incident reporting scheme (Braun, 2002). Conventional dynamic models incorporate several key notations useful for examining systems failures. Table 2.1 summarizes the symbols used in these dynamic models. In particular, the system boundary notation in dynamic model representation is effective for preventing the introduction of side effects by reinforcing incorrect countermeasures. Symbol R or B can be combined with IC or UC; for example, BIC stands for a balancing intended consequences loop. The "+" sign indicates that an increase or decrease in state 1 causes an increase or decrease, respectively, in state 2. The "-"sign indicates that an increase (decrease) in state 1 causes a decrease (increase) in state 2. The problem and side effect archetypes clarify the leverage points of problems when introducing countermeasures.

Table 2.1 Symbols used in dynamic models

Symbol/Notation	**Feature**
R	Reinforcing loop
B	Balancing loop
=	Time delay of an effect
—	System boundary
IC	Intended consequences (combination with R or B)
UC	Unintended consequences (combination with R or B)
+	Positive feedback loop
-	Negative feedback loop
Problem	Problem type of dynamic model
Side effect	Side effect type of dynamic model
Solution	Solution type of dynamic model

4. SAFETY ARCHETYPES IN ENGINEERING SYSTEM FAILURES

4.1 Overview of safety archetypes and its behavior through time

There are three strands of problem archetypes and solutions: (1) a system failure archetype for all failure classes; (2) an archetype of misunderstanding Class 2 and 3 failures as Class 1; and (3) an archetype of misunderstanding failures of Class 1 as Class 2 or 3. We exclude the third strand because all engineering system failures have technical components, so Class 1 is always within the scope of analysis. Figure 4.1 illustrates the transition of engineering safety archetypes through time. Both first and second strands have solution archetypes derived from single loop learning (third column in Fig. 4.1). These solution archetypes seem to work within a short period of time but then gradually introduce various side effects (fourth column in Fig. 4.1). The solution archetypes from double loop learning access the real root causes in order to enhance engineering safety (fifth column in Fig. 4.1). Sections 4.2 to 4.11 explain each scenario of the dynamic model shown in Fig. 4.1. Stage I, II and VI in Fig. 4.1 are explained in Table 4.1.

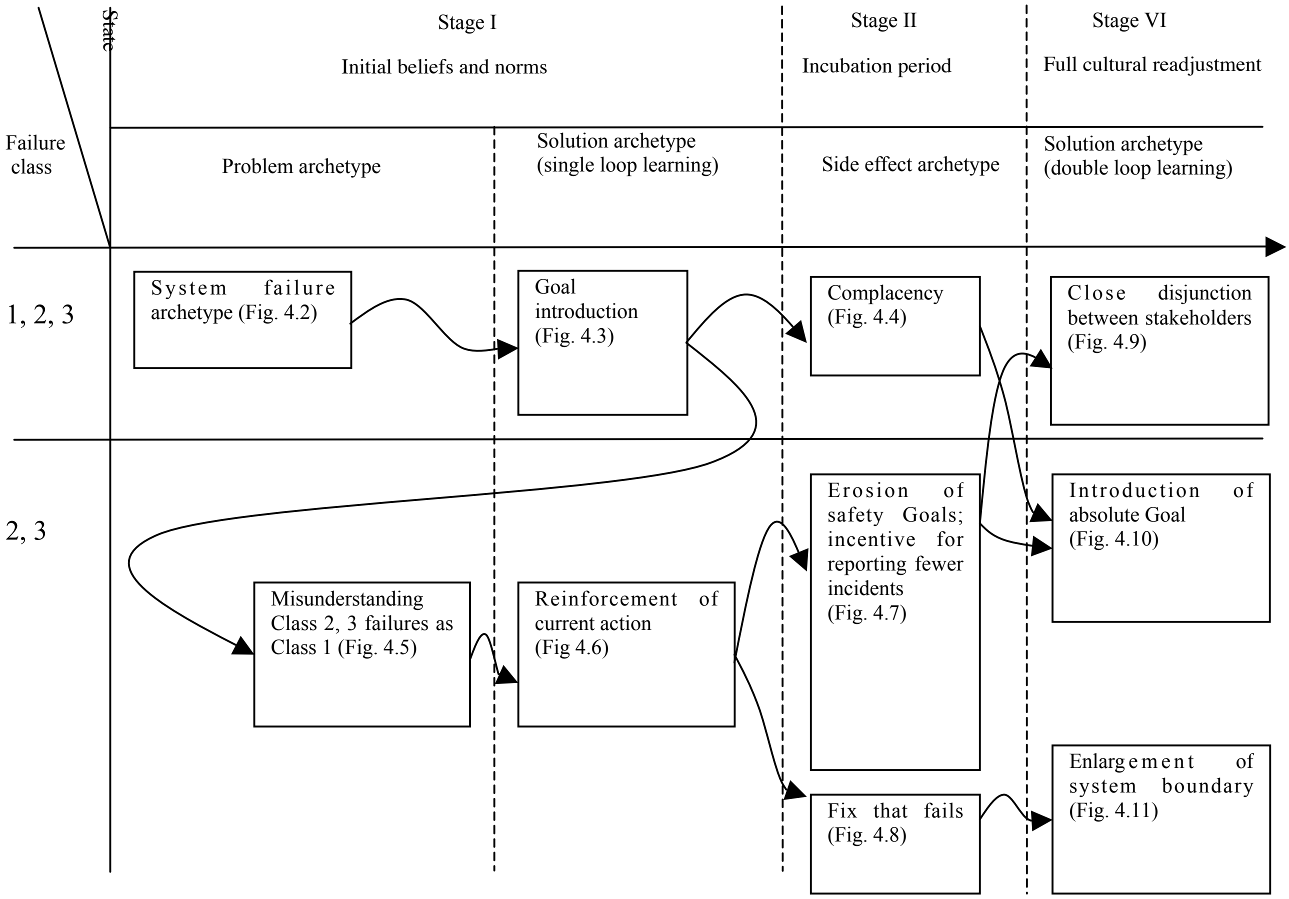
State
Stage I
Initial beliefs and norms
Stage II
Incubation period
Stage VI
Full cultural readjustment
Failure class
Problem archetype
Solution archetype (single loop learning)
Side effect archetype
Solution archetype (double loop learning)
1, 2, 3
2, 3
System failure archetype (Fig. 4.2)
Goal introduction (Fig. 4.3)
Complacency (Fig. 4.4)
Close disjunction between stakeholders (Fig. 4.9)
Erosion of safety Goals; incentive for reporting fewer incidents (Fig. 4.7)
Introduction of absolute Goal (Fig. 4.10)
Misunderstanding Class 2, 3 failures as Class 1 (Fig. 4.5)
Reinforcement of current action (Fig 4.6)
Enlargement of system boundary (Fig. 4.11)
Fix that fails (Fig. 4.8)

Fig 4.1 Problem and solution archetypes in engineering system failures through time

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 4.1 shows the feature of each stage and its relation between six stages, Failure Classes and Safety Archetypes explained above.

Table 4.1 six stages of development system failures and its relation to safety archetypes

State of development	Feature	Failure Class	Safety Archetype
Stage I Initial beliefs and norms	Failure to comply with existing regulations Class1	Class1	System Failure Archetype (Fig.4.2)
			Goal introduction (Fig.4.3)
			Reinforcement of current action (Fig.4.6)
Stage II	Events unnoticed or misunderstood because of	Class3	Complacency (Fig.4.4)

Incubation period	misunderstood because of erroneous assumptions	Class2 and 3	Misunderstanding Class 2, 3 failure as Class 1 (Fig.4.5)
	Events unnoticed or misunderstood because of difficulties of handling information in complex situations	Class2	Fix that fail (Fig.4.8)
	Effective violation of precautions passing unnoticed because of 'cultural –lag' in formal precautions	Class1 and 3	Erosion of safety Goals (Fig.4.7)
	Events unnoticed or misunderstood because of a reluctance to fear the worst outcome	Class3	Incentive to fewer incidents (Fig.4.7)
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	Close disjunction between stakeholders (Fig.4.9)
			Introduction of absolute Goal (Fig.4.10)
			Enlargement of system boundary (Fig.4.11)

4.2 System failure archetype (problem)

Figure 4.2 illustrates the system failure archetype. A system failure requires a counteraction that acts on the root cause and mitigates a Class 1 failure in the end. This is a very simple scenario, because the failure and its causes are within the system. The achievements of this archetype, however, saturate at some point of the time because of the BIC loop. If the saturation point of performance is well beyond the target or goal, this might not be an issue. Otherwise, we need another solution for this relative achievement situation in which the intended goal is not achieved.

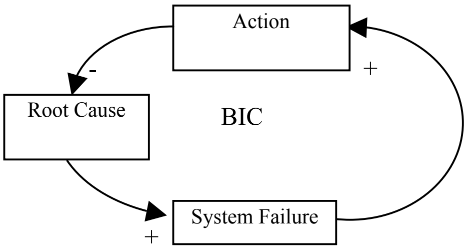

Fig 4.2 System failure archetype (problem)

4.3 System failure archetype (solution)

Figure 4.3 illustrates the system failure solution archetype. A simple solution for system failure is to introduce a goal, and compare it with the current status, and adjust the action. This introduces a reinforcing action, until the goal has been achieved. This RIC loop breaks the balanced situation of the circle on the left side of the figure. This is a very simple scenario for the solution archetype of system failure. It is a typical example of single loop learning and is a predominant feature of current troubleshooting technologies.

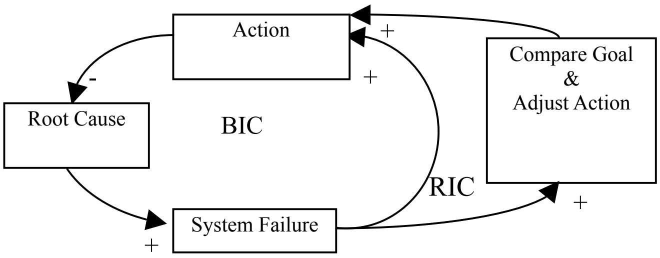

Fig 4.3 System failure archetype (solution)

4.4 Complacency archetype (side effect)

This problem archetype (Fig. 4.4) is the side effect of the system failure archetype (solution). The action loop of the system failure archetype (solution) continues for some time. This increases awareness of safety within the system boundary, which in turn generates oversight and finally leads to system failure again. This relative achievement situation explains why system failures repeat over a longer time span.

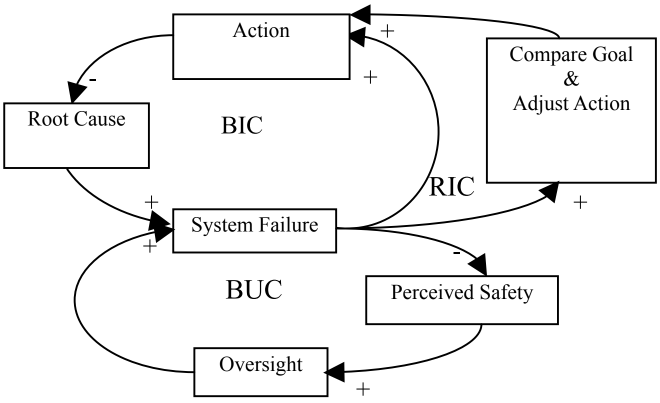

Fig 4.4 Complacency archetype (side effect)

4.5 Misunderstanding Class 2 or 3 failure as Class 1 archetype (problem)

This archetype (Fig. 4.5) explains why system failure repeats after introducing a quick fix or inappropriate fix. It might reduce system failure in the short term and then gradually saturate the effect at a level below the organization's goal. The BIC loop becomes open, with no further effect from the quick fix. The lower BIC loop in Fig 4.5 becomes open as a result of misunderstanding the system failure class and introducing no essential effects to solve the original problem.

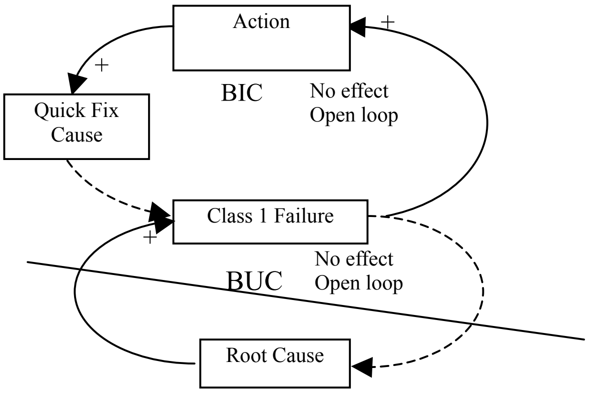

Fig 4.5 Misunderstanding system failure archetype (problem)

4.6 Misunderstanding Class 2 or 3 failure as Class 1 archetype (solution)

This is a single loop learning scenario (Fig. 4.6) that introduces a reinforcing action based on the deviation from a predetermined goal. The RIC action to improve the situation escalates to the introduction of further quick fixes, only to repeat a similar scenario. The RIC action causes various side effects, including erosion of safety goals and an incentive to report fewer incidents. These side effects are hard to detect because the performance malfunction alarm becomes mute and management review can oversee these effects only by checking quantitative performance. This explains why such system improvement is bound to fail, as van Gigch (van Gigch, 1991) points out. In this relative achievement situation, a real root cause outside the system boundary should be dealt with.

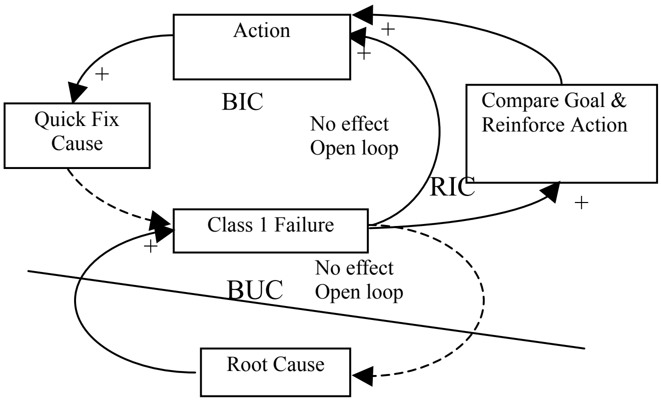

Fig 4.6 Reinforcing current action archetype (solution)

4.7 Erosion of safety goals and incentive to report fewer incidents (side effect)

This side effect is introduced by an RIC loop that becomes tight without any further success in reducing system failures (Fig. 4.7). Increased pressure to achieve a goal emerges from the BUC loop by shifting the goal (i.e., lowering it) and hiding the real state of quality or safety from management. In this relative achievement scenario, a manager who stays within the system boundary has difficulty detecting the real state of achievement. This is why many Japanese manufacturers have a slogan of "3R-ism," which ask managers to see if they have identified a problem at a "real site," confirmed it with "real objects," and discussed it with a "real person in charge," before taking any action.

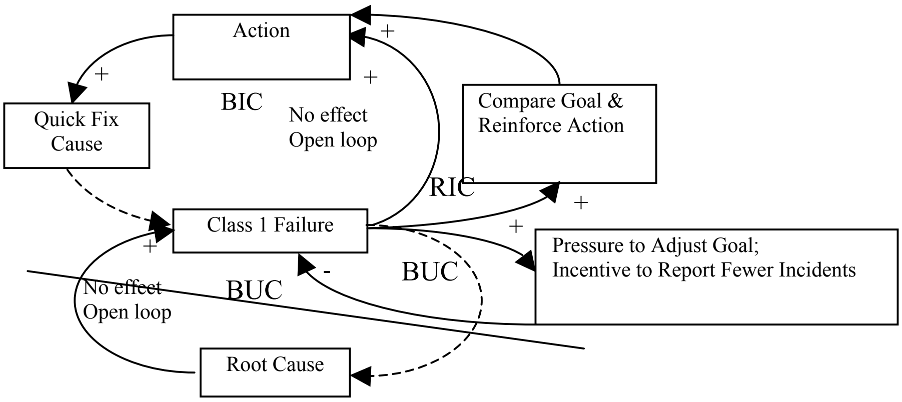

Fig 4.7 Erosion of safety goals and incentive to report fewer incidents (side effect)

4.8 Fix that fails archetype (side effect)

Figure 4.8 illustrates a typical example of local optimization. The action taken for the root cause is a short-term solution to the problem and introduces delayed, unintended consequences outside the system boundary, which introduces a failure of Class 2 or 3. For example, an operations manager might shift resources from a proactive task team to a reactive task team because of a rapid increase in system failures, which would only cause the RUC loop to further increase the occurrence of system failures. This out-of-control situation can only be managed at the expense of others and damages the organization in a longer perspective.

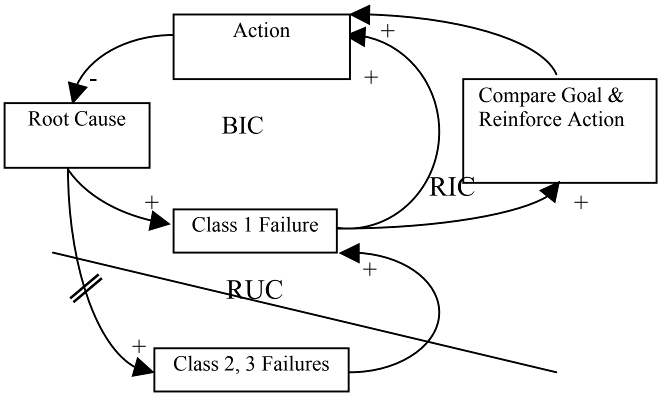

Fig 4.8 Fix that fails archetype (side effect)

4.9 Double loop learning for Class 2 failure archetype (solution)

It is necessary to focus on the possibilities of relative achievement or the side effects of a quick fix. A tacit assumption of stakeholder disjunction should be accommodated through debate to close the responsibility gap. Figure 4.9 shows this solution for the scenario shown in Fig. 4.5, misunderstanding system failure archetype (problem).

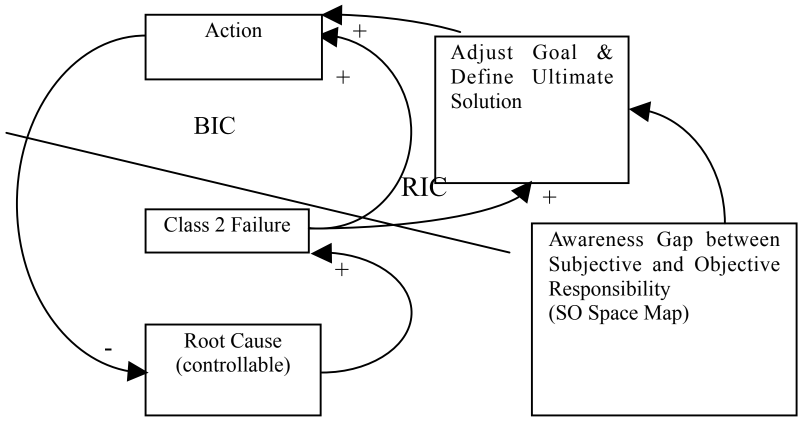
Action
+
+
Adjust Goal & Define Ultimate Solution
BIC
RIC
+
Class 2 Failure
Awareness Gap between Subjective and Objective Responsibility (SO Space Map)
+
-
Root Cause (controllable)

Fig 4.9 Double loop learning for Class 2 failure (solution)

4.10 Double loop learning for class 3 failure archetype (solution)

As explained in section 2, the speed of technology advancement and growth of complexity are unpredictable. Therefore, a current goal could later become obsolete. This could be a real root cause of system failure, with no party responsible for the failure. In other words, the system failure emerges through no one's fault. This kind of failure can be avoided by periodically monitoring goal achievement and benchmarking competitors. Figure 4.10 illustrates this scenario.

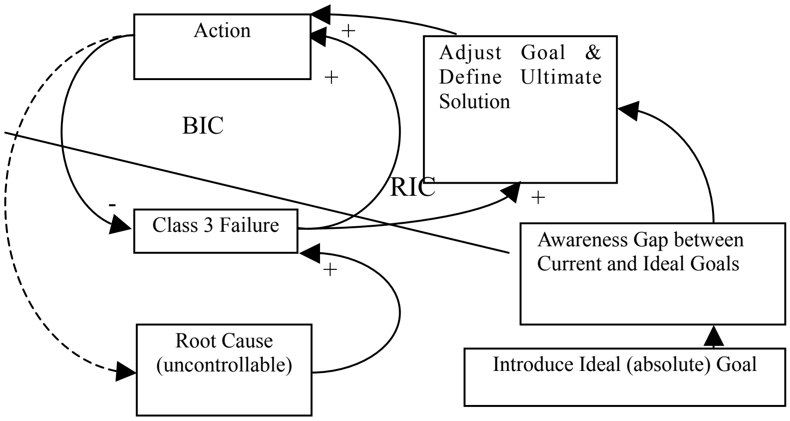
Action
+
+
Adjust Goal & Define Ultimate Solution
BIC
RIC
-
+
Class 3 Failure
Awareness Gap between Current and Ideal Goals
+
Root Cause (uncontrollable)
Introduce Ideal (absolute) Goal

Fig 4.10 Double loop learning for Class 3 failure (solution)

4.11 Double loop solution for fix that fails archetype (solution)

The solution of this archetype is to raise the viewpoint of the problem (Fig. 4.11). Class 2 and 3 failures become Class 1 if the presumed system boundary is enlarged. In addition, a solution link between groups would change the RUC loop to a BIC loop, which would be beneficial for achieving both groups' goals.

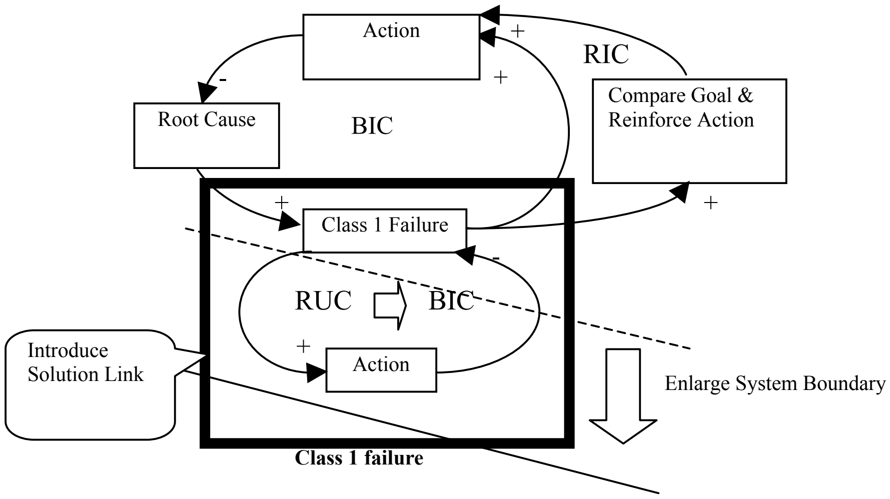

Fig 4.11 Double loop learning for fix that fails archetype (solution)

5. ACTUAL APPLICATION SCENARIO APPLYING THE DYNAMIC MODEL AS DOUBLE LOOP LEARNING

Although above safety archetype has the capability to examine the dynamic aspects of system failures, a longer perspective like the “History as Cause” mentioned by NASA (section 4) should be intentionally employed in real application of the dynamic model. Reason (Reason, 1997) explains the organizational life span between protection and catastrophe. The lifespan of a hypothetical organization through production-protection space (Fig. 5.1) explains why organizational accidents repeat, with this history ending in catastrophe. This is why the side effects of dynamic movement should be confirmed.

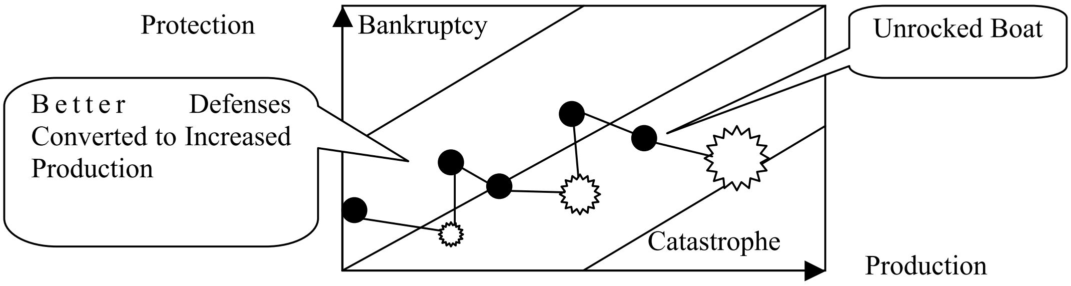

Fig 5.1 Lifespan of a hypothetical organization through production-protection space

We need to introduce OP matrix in order to confirm side effect properly. OP stands for objective and problem. The OP matrix is used to reveal disjunctions between objectives and problems in order to verify that current objectives fully encompass past system failures (Fig. 5.2). The first quadrant, where (P, O) = (OK, OK), is the normal situation, because a goal has been achieved and there is no repetition of similar problems. The second quadrant, where (P, O) = (NG, OK), might indicate a disjunction between stakeholders. This could be a manifestation of a problem for which no one has responsibility. The third quadrant, where (P, O) = (OK, NG), might indicate a system failure that is not yet fully manifested. A goal might have to be altered in order to capture the real state of problem repetition. In the fourth quadrant, where (P, O) = (NG, NG), a hard paradigm approach might be effective. Management

Malfunction (i.e., “extinct by instinct”) can cause this situation. The best practical scenario for applying the OP Matrix is during periodic management review.

Problem repetition

Goal achievement

P O	OK	NG
OK	Normal	Disjunction between stakeholders
NG	Sign of system failure	Hard approach effective

Fig 5.2 OP matrix (objective-problem)

We can use the OP matrix to understand why a system failure repeats or a fix does not work in the long term. Figure 5.3 shows a vicious circle of repeating system failures. This indicates that management review of engineering safety should be careful even if the current state is in the first quadrant, where (P, O) = (OK, OK). The state can transfer to the fourth quadrant, where (P, O) = (NG, NG), through introduction of the complacency archetype (Fig. 4.4). If the misunderstanding system failure archetype (Fig. 4.5) happens in the third quadrant, the situation transfers to another quadrant: the third, where (P, O) = (OK, NG), for the erosion of safety goal archetype; or the second, where (P, O) = (NG, OK), for the incentive to report fewer incidents archetype by reinforcing the current action archetype (Fig. 4.6). This is followed by a further transfer back to the first quadrant, giving management a false impression that safety goals have been achieved. This is another explanation of organizational navigation leading to catastrophe (Fig. 5.1), like the normalized deviation effects in the space shuttle disasters.

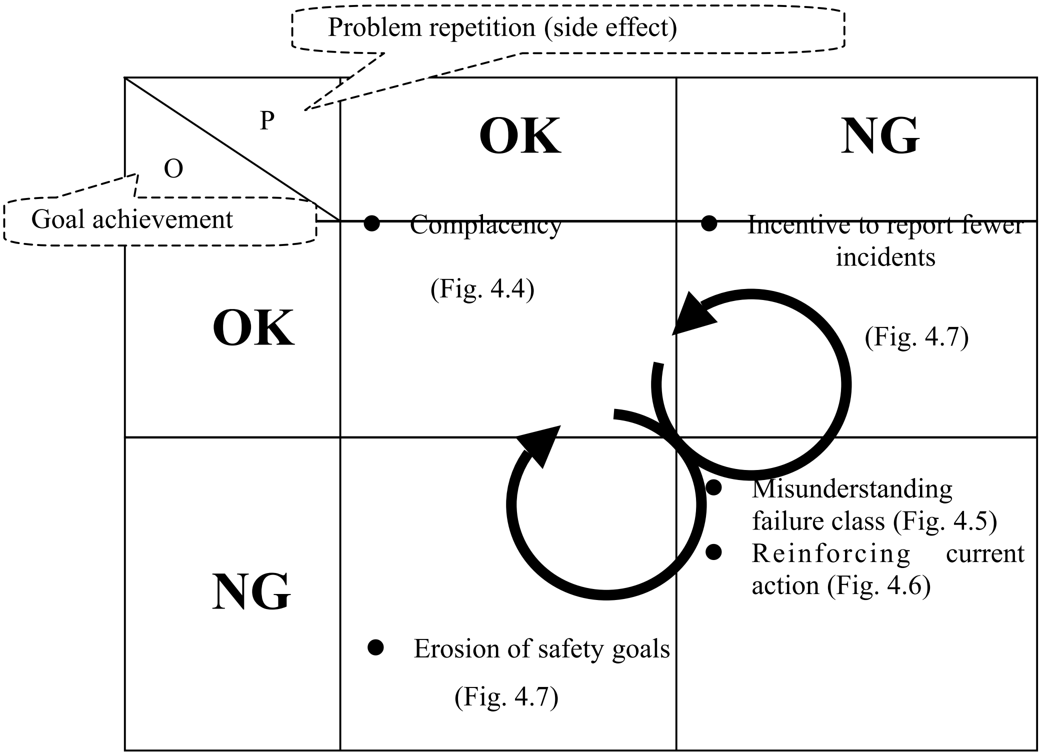
Problem repetition (side effect)
P
O
Goal achievement
OK
NG
OK
NG
Complacency
(Fig. 4.4)
Incentive to report fewer incidents
(Fig. 4.7)
Misunderstanding failure class (Fig. 4.5)
Reinforcing current action (Fig. 4.6)
Erosion of safety goals
(Fig. 4.7)

Fig 5.3 Vicious circles indicating repeated system failures.

OP matrix helps to identify the long-term dynamic aspects of system failure and promotes double loop learning, as it changes the action model.

6. APPLICATION EXAMPLE: SERVER NOISE PROBLEM - DESIGN FAILURE OR INSTALLATION ERROR?

In this example, a PC server user complains about the noise of running such servers in an office environment. It takes time for the PC server manufacturer to modify the noise design specification to conform to office utilization of the server. At the first stage, this is not treated as the designers' fault, because there had already been a design norm for the noise level, and the server noise conformed to this predetermined level. The problem, however, is that the designers' assumption of operation in a machine room environment was not communicated to customers. At the first stage, this is treated as a Class 1 system failure without further improvement in the situation. This introduces the side effects of erosion of goals and an incentive to report fewer incidents. If this system failure will be treated as Class 3, an evolutional malfunction, because the goals of the designers and the installer (or end user) have differed in time.

Countermeasures for only Systems 1 to 3 are inadequate, because the root cause resides in a soft system paradigm (Checkland, 1981; Checkland, Holwell, 1998), and Systems 4 and 5 should be modified to alter the design norm of the server noise level. Raising the countermeasure into System 4 and 5 is important for reflecting the noise level specifications of other servers (for example, a UNIX server, as opposed to a PC server). This will prevent other server problems by also modifying the design norm for the UNIX server. Figure 6.2 shows the differences in prevention level between reality, modeling, and meta modeling. These differences are also confirmed by using the dynamic model. If the noise problem is treated as a Class 1 failure, the side effects of erosion of safety goals and an incentive to report fewer incidents will appear in the long term (Fig. 6.3). This is the state of "normalized deviance" as mentioned above. We need to reinterpret this Class 1 failure as Class 3 failure turning hindsight to foresight. Figure 6.4 illustrates that treating the noise problem as Class 3 will lead to essential resolution. This dynamic model is quite powerful, and it is easy to understand that problems at the levels of Systems 4 and 5 should be escalated to the management layer. This avoids unnecessary cost in reaching a final decision to lower the noise level norm at the design phase.

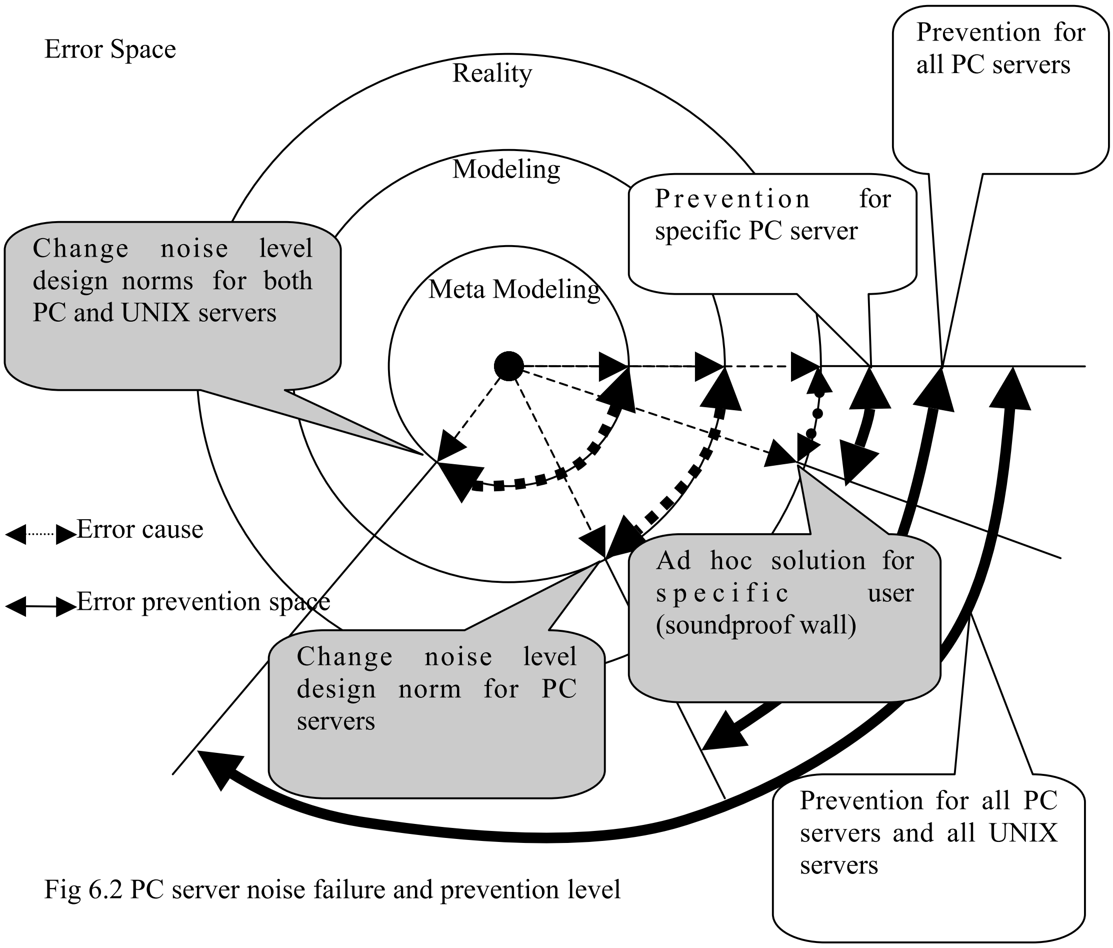
Error Space
Reality
Modeling
Meta Modeling
Prevention for all PC servers
Prevention for specific PC server
Change noise level design norms for both PC and UNIX servers
Error cause
Error prevention space
Ad hoc solution for specific user (soundproof wall)
Change noise level design norm for PC servers
Prevention for all PC servers and all UNIX servers

Fig 6.2 PC server noise failure and prevention level

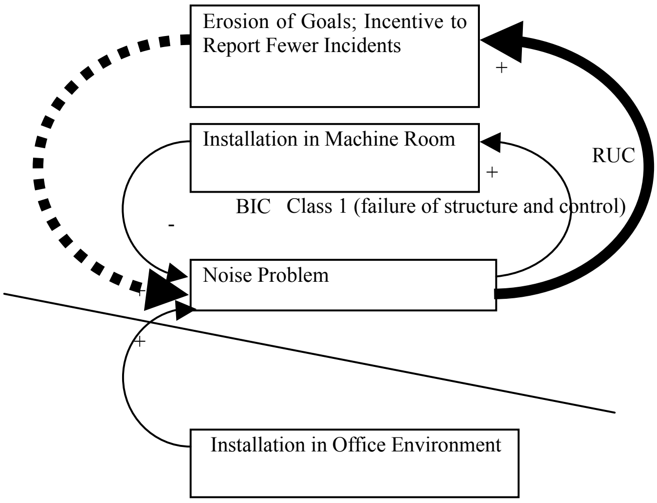
Erosion of Goals; Incentive to Report Fewer Incidents
+
Installation in Machine Room
+
RUC
BIC Class 1 (failure of structure and control)
-
Noise Problem
+
+
Installation in Office Environment

Fig 6.3 Class 1: evolutional noise failure (with erosion of goals or incident reporting)

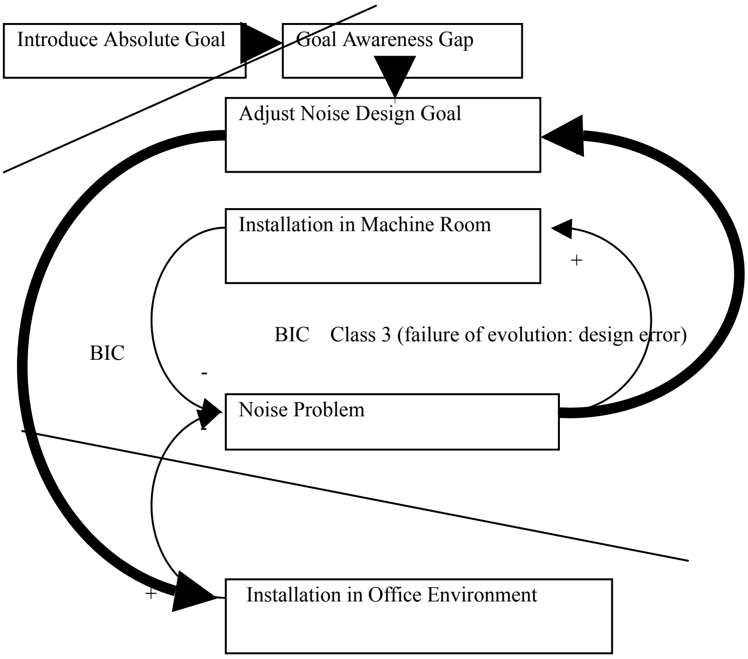
Introduce Absolute Goal
Goal Awareness Gap
Adjust Noise Design Goal
Installation in Machine Room
+
BIC Class 3 (failure of evolution: design error)
BIC
-
Noise Problem
Installation in Office Environment
+

Fig 6.4 Class 3: evolutional noise failure

In this example, the mental model is changed through introduction of an absolute (ideal) goal by benchmarking competitors. The operating norm is changed by changing the design goal (i.e., the noise level), and the current process is changed by changing the operating norm. Dynamic transition of turner's six stages is shown with OP matrix in Fig 6.5. Double loop learning is achieved through incubation period with some side effects (i.e. misunderstanding failure class, reinforcing current action and incentive to report fewer incidents).

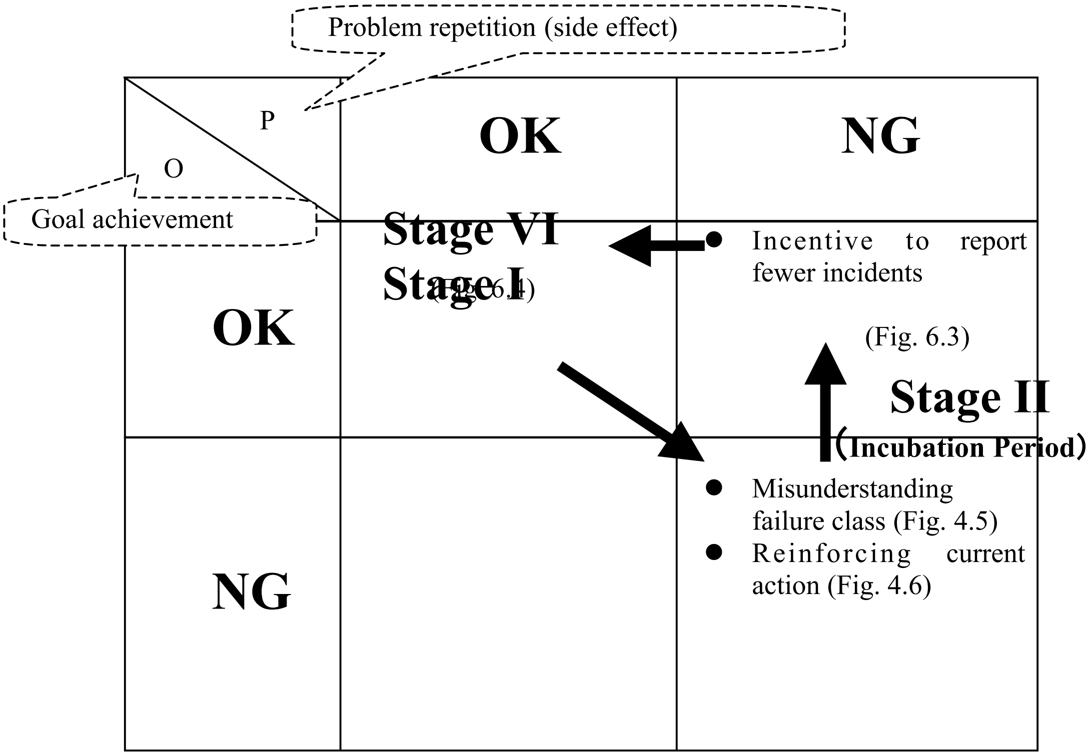
Problem repetition (side effect)
P
O
Goal achievement
OK
NG
OK
NG
Stage VI
Stage I
(Fig. 6.4)
Incentive to report fewer incidents
(Fig. 6.3)
Stage II
(Incubation Period)
Misunderstanding failure class (Fig. 4.5)
Reinforcing current action (Fig. 4.6)

Fig 6.5 Learning cycle through incubation period

7. CONCLUSION

The concept of an inquiring system (IS), introduced by van Gigch (van Gigch, 1991), describes how the black-box concept is elaborated as a decision-making process. Epistemology consists of the thinking and reasoning processes by which an IS acquires and guarantees its knowledge. Furthermore, epistemology converts evidence into knowledge and problems into solutions, designs, or decisions. Learning at the meta level modifies mental models, while at the model level, it changes the desired goal and the current operating norm, and at the reality level, it changes the operation. The outcome of double loop learning is an epistemology of experience. The example in section 6 demonstrates that the proposed meta methodology can actually promote double loop learning. The epistemologies acquired through this example turning hindsight to foresight are as follows.

i) Enlarge system boundary as much as possible to convert a system failure of Class 2 or 3 to Class 1.
ii) Sense “normalized deviance” state and respond Class 3 failure.
iii) Close stakeholder disjunctions to reduce Class 2 failures.
iv) Set absolute goals to reduce Class 3 failures.
The example also shows the efficacy of this methodology. If the level of countermeasures is raised up to the meta model layer, the effect of the countermeasures is increased; otherwise, similar problems would repeatedly occur sometime later. The predominant methodologies are only effective when a system failure is Class 1. Management pressure on Systems 1 to 3 as Class 1 failures causes various side effects and damages the organization in the long term. Reflective

recognition of system failures by using this dynamic model and its related tools can show the way to establish engineering safety even in uncertain, rapidly changing environments.

REFERENCES

Argyris C., Schoen D. 1996, Organizational Learning II, Addison Wesley: Mass..

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.

Braun W. 2002. The system archetypes. http://www.uni-klu.ac.at/~gossimit/pap/sd/wb_sysarch.pdf [23 Dec 2007].

Checkland P. 1981. System thinking, system practice. John Wiley & Sons: UK.

Checkland P., Holwell S. 1998. Information, Systems and Information Systems making sense of the field. John Wiley & Sons: UK.

Leveson N. 2004. A new accident model for engineering safer systems. Safety Science vol. 42, issue 4: 237-270.

Nakamura T., Kijima K. 2007. Meta system methodology to prevent system failures. Proceeding of ISSS 2007.

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

Morgan G. 1986. Images of Organization. Sage Publications: California.

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 accidents. Ashgate Publishing limited: 3-5.

Senge P. 1990. The Fifth Discipline: The Art and Practice of the Learning Organization, 1st edition. New York.

The Columbia Accident Investigation Board Report Chapter 6. http://history.nasa.gov/columbia/CAIB_reportindex.html [23 Dec 2007]. chapter 6:130, chapter 8: 185.

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

van Gigch J. P. 1986. Modeling, Metamodeling, and Taxonomy of System Failures. IEEE trans. on reliability, vol. R-35, no. 2, 1986 June: 131-136.

van Gigch J. P. 1991. System design Modeling and Metamodeling. Plenum: New York.

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

Vaughan D. 1997. The Challenger Launch Decision: Risky Technology, Culture, and Deviance at Nasa. Univ of Chicago Pr. London