3

System of System Failure: Meta Methodology to Prevent System Failures

Takafumi Nakamura[1] and Kyoichi Kijima[2]
[1]*Fujitsu Fsas Inc.,*
[2]*Tokyo Institute of Technology,*
Japan

1. Introduction

The purpose of this chapter is to propose a meta methodology 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, 2007, 2008a). 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. There are many examples of similar system failures repeating and of negative side effects created by quick fixes. Introducing safety redundant mechanisms does little to reduce human errors. As pointed out by Perrow (1999, p. 260), the more redundancy is used to promote safety, the greater the chance of spurious actuation; "redundancy is not always the correct design option to use." While instrumentation is being improved to enable operators to run their operations more efficiently and certainly with greater ease, the risk would seem to remain about the same. 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). This indicates that we need a meta methodology 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 chapter, we propose a meta methodology called System of System Failures (SOSF), along with a system diagnostic failure flow, in order to overcome the current methodologies' shortcomings. We also demonstrate this meta methodology's efficacy through an application in IT engineering.

In the next section, we explain the current troubleshooting techniques' features and limitations with respect to certain aspects of system failures. Section 3 describes the three key features required in order to overcome these limitations, as well as SOSF, which actually overcomes the limitations. In section 4, we propose the actual application scenario that fully utilizes SOSF to promote double loop learning, or total system intervention for system failure (TSI for SF). The SOSF and related methodologies are used in the course of the subsequent discussion and debate to agree upon who is responsible for the failure and to identify the preventative measures to be applied. In section 5, an application example in information and communication technologies engineering demonstrates that using the proposed "TSI for SF" helps prevent future system failures by learning from previous system failures, followed by a concluding discussion of a efficacy of the SOSF and three actions were identified for preventing further system failures: close the gap between the stakeholders, introduce absolute goals and enlarge system boundary.

2. Limitations of current troubleshooting techniques

The predominant technology of current ICT troubleshooting is based on a predefined goal-seeking model. van Gigch (1991) points out the main shortcomings of system improvement in this model, as follows. (1) Engineers look for causes of malfunctions within the system boundary. The rationale of system improvement tends to justify systems as ends in themselves, without considering that a system exists only to satisfy the requirements of larger systems in which it is included. (2) Engineers seek to restore systems back to normal. A lasting solution cannot result from an improvement in the operation of a present system. An improvement in operations is not a lasting improvement. (3) Engineers tend to hold incorrect, obsolete assumptions and goals. It is not difficult to find organizations in which the formulation of assumptions and goals has not been explicit. Fostering system improvement in this context is senseless. (4) Engineers act as "planner followers" rather than as "planner leaders." Another manifestation of the problem of holding incorrect assumptions and pursuing the wrong goals can be traced to different concepts of planning and of the planner's role. In the context of system design, the planner must be a planner leader, planning to influence trends, instead of a planner follower, planning to satisfy trends.

This chapter focuses on system failure aspects that current methodologies cannot manage properly in the sense pointed out by van Gigch. To summarize, these aspects are soft, systemic, emergent, and dynamic; i.e., they accommodate multiple stakeholders' worldviews (Checkland, 1981; Checkland & Holwell, 1997).

Technology is changing faster than engineering technology can treat system failures. The growing increase in CPU power versus price is well known in the form of Moore's law. Moreover, the numbers of stakeholders in computer systems is getting bigger and bigger. For computer architects, the stakeholders should encompass clients of clients (i.e., end users) in order to satisfy ICT system owner's requirements. ICT system provider should focus on the dynamic aspects of end users and ICT system owners (e.g., through capacity planning of web banking system design), as well as on computer components (HDDs, CPUs, etc.) supplied by various vendors in order to implement synthesized functions. The environmental changes surrounding ICT systems, in terms of speed and complexity, are increasing over time. The problem is that once a system failure happens under these circumstances, it is extremely difficult to identify the real root cause. Most troubleshooting methodologies view system failures as resulting from a sequence of events. Furthermore, they focus mainly on the technical aspects of system failures. These models are only suitable for a relatively simple system with unitary participants from a technical perspective.

The following four key features are commonly pointed out for the current troubleshooting methodologies surrounding ICT system environments. 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 (1999) argues that the conventional engineering approach to ensure safety – building in more warnings and safeguards – fails because system complexity makes failures inevitable.

1. Current methodologies are technically well established (e.g., ISO and IEC standards) but are not always helpful for understanding the real implications of countermeasures and whether they are real solutions or merely tentative fixes from outside the technical arena. Moreover, most methodologies are based on a reductionist worldview.
2. The current troubleshooting mainstream applies cause-effect analysis (or event chain analysis) to find out real root causes. Forward sequences (as in FMEA or event trees) or backward sequences (as in fault trees) are often employed (IEC 60812 (2006), IEC 61025 (2006)). Toyota has a corporate slogan suggesting to "ask why five times" to reach root causes. This promotes finding "what" in order to seek counter measures to the problem. This approach, however, tends to become a victim-finding tool for blaming a specific person or group rather than finding a real root cause.
3. The enormous speed of technological advance causes various misunderstandings between ICT system stakeholders. This responsibility disjunction cannot be managed properly with current methodologies.
4. Improvement of the deviation from operating norm is bound to fail, as van Gigch (1991) points out that the treatment of system problems by improving the operation of existing systems is bound to fail. Current troubleshooting methodologies focus on the following main problems:

- The system does not meet its established goals.
- The system does not yield predicted results.
- The system does not operate as initially intended.

The basic assumption of improvement is that the goal and operating norm are static and predetermined at the design phase and are based on hard systems thinking.

The above four features hinder examination of system failures from a holistic viewpoint, making it impossible to manage the soft, systemic, emergent, and dynamic aspects of system failures.

3. Double loop learning and System of System Failures (SOSF)

3.1 Double loop learning and three key success factors for new methodology

To overcome the current methodological shortcomings discussed above, we need to promote double loop learning. The most important key success factor is the ability to ask a question with respect to a current operating norm (i.e., a mental model). Skill in double loop learning should enable people to question basic assumptions, which leads to modification of their mental models (Fig. 1) to create action producing desired goals, rather than simply modification of their actions under current mental models (Argyris & Schoen, 1996; Morgan, 1986; Senge, 1990).

Double loop learning should influence all three layers listed in Table 1: reality is for changing actions, model is for changing desired goals, and meta is for modifying mental models. Figure 1 explains single and double loop learning in a multi-stakeholder environment based on a double loop learning model (Morgan, 1986). The dotted line in Fig.

1 indicates one specific stakeholder for achieving a goal. The one stakeholder alone is not enough to overcome current methodological shortcomings. We should thus expand double loop learning to account for a multi-stakeholder situation. Under this situation, there are three key success factors for overcoming current methodological shortcomings. First, there should be common language among the stakeholders' mental models (i.e., the mental model box in each stakeholder's domain in Fig. 1). Otherwise, the failures caused by stakeholders' mental model gaps will not be resolved effectively. Second, there should be a meta methodology (i.e., the meta model box in Fig. 1) to promote double loop learning. This meta methodology should be unique between stakeholders; otherwise, the mutually exclusive and collectively exhaustive (MECE) nature of countermeasures is hard to achieve. Therefore, there is only one meta model box in Fig. 1, and it is shared among stakeholders. Third, there should be failure classes based on the origin of a failure. This is essential to ensure the efficacy of countermeasures. There are three origins of system failures: i) the mental model, ii) a mental model gap between stakeholders, and iii) the meta model. These three origins correspond to failure classes 1 (failure of deviance), 2 (failure of interface), and 3 (failure of foresight), respectively, as indicated in Fig. 1. The following explains the three key success factors in detail.

1. We should have a common language for understanding system failures. It is vital to examine system failures from various perspectives. System safety can be achieved through the actions of various stakeholders. One such common language was developed by van Gigch (1986) for taxonomy of system failures. There are six categories of system failures, namely, failures of i) technology, ii) behavior, iii) structure, iv) regulation, v) rationality, and vi) evolution.
2. We should have a meta methodology to ensure that countermeasures are correct and essential rather than just quick fixes that introduce long-term side effects. To redress system malfunctions or a system failure, it is necessary first to translate specific failure events into a model world in order to appraise the nature of reality holistically, then to discuss the system failure's model in the modeling phase (i.e., metamodeling) in order to investigate why the failure happened, what the countermeasures are, and what should be learned in the organizational process so as to avoid further occurrence of the failure. Kickert (1980) explained an organizational structure model corresponding to the organizational purpose and breaking the organizational structure down into three layers: the aspect system, subsystem, and phase system. These layers relate to "what," "who," and "when," respectively. Beer's VSM model (Beer, 1979; 1980) rectifies the organizational process. Systems 1 to 3 are the operational level, and systems 4 and 5 are the meta level for deciding the operating norm through communication outside the system environment. There are hierarchical similarities between Kickert's and Beer's models, as follows:
    - Systems 1 to 3 correspond to the phase system managing "when." These levels ensure internal harmony and maintain internal homeostasis. Systems 1, 2, and 3 represent when an operation should be done, how it is coordinated, and how to maintain corporate management, respectively.
    - System 4 for strategic corporate management corresponds to the subsystem managing "who." This level integrates internal and external inputs in order to chart enterprise strategies (i.e., external homeostasis) and clarifies who should be responsible for those strategies.

- System 5 for normative corporate management corresponds to the aspect system managing "what." This level formulates long-term policies (i.e., planning and foresight) and decides what should be done.

Kickert's organizational model and Beer's VSM model both decompose organization into three layers: reality (i.e., operation), model (i.e., adaptation), and meta (i.e., evolution). The reality and model layers seek to answer "how," and the meta layer seeks to answer "what." This differentiation is crucial to ensure the efficacy of countermeasures. Table 1 summarizes the relations between the organizational structure (Kickert, 1980) and VSM (Beer, 1979; 1980) models.

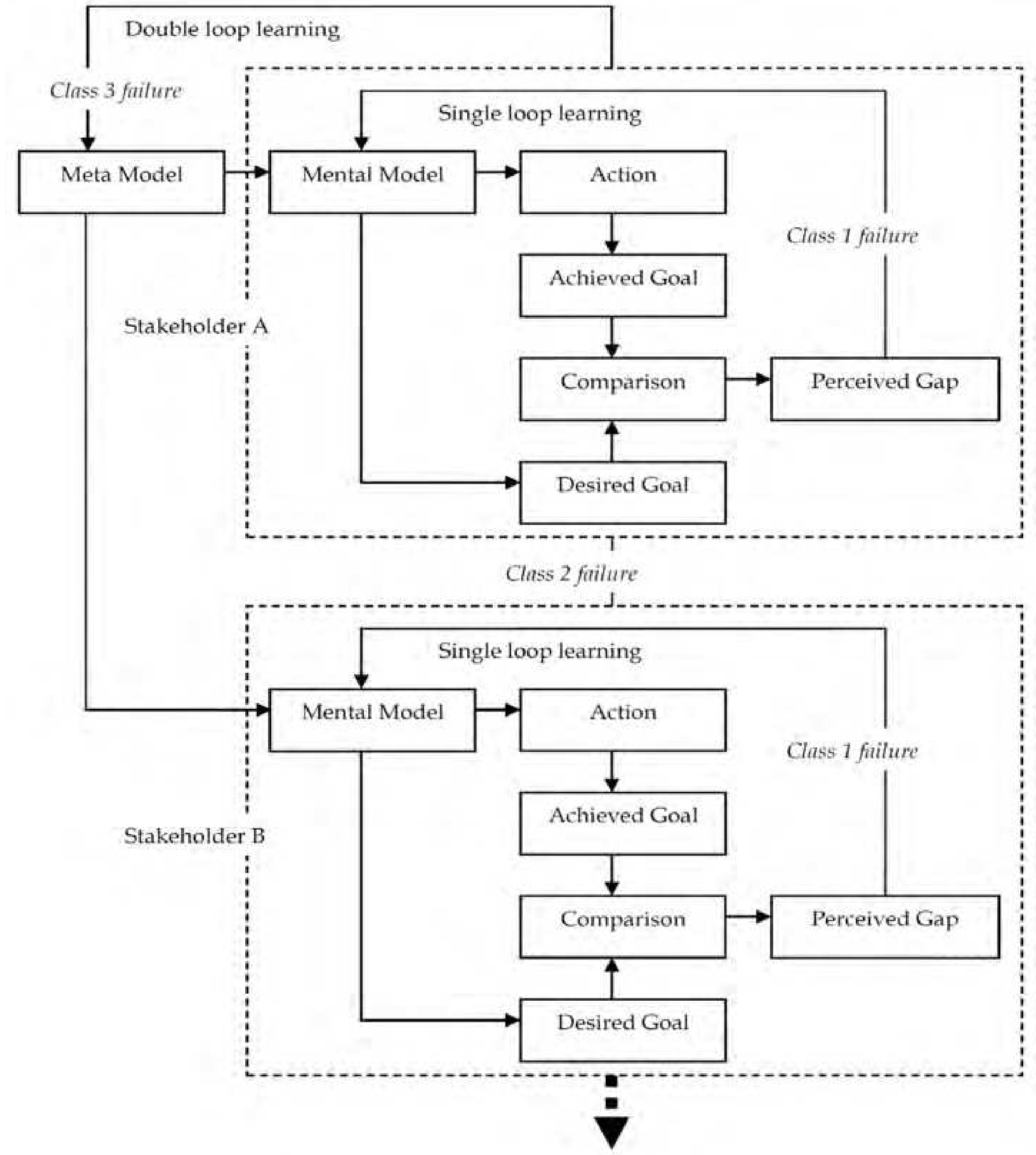


Fig. 1. Single and double loop learning under a multi-stakeholder environment.

	Organization structure	Objective	VSM
Meta	Aspect system: What	Mental model	System 5
Model	Subsystem: Who	Operating norm	System 4
Reality	Phase system: When	Operation	Systems 1-3

Table 1. Relations between the organization structure (Kickert) and VSM (Beer) models.

3. We should be able to specify three failure classes in order to avoid the dynamic aspects of system failures (i.e., erosion of safety goals over time). These failure classes should intentionally be identified in conjunction with the VSM model. 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 to prevent 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 resulting from 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 and Kickert's model serve well for understanding the real root causes.

In a stable environment, control of activities and maintenance of their safety through 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. Section 3.4 thus describes a dynamic model explaining why fixing failures sometimes introduces unintended side effects and how dynamic understanding contributes to introducing countermeasures that are ultimately more effective.

3.2 System of System Failures (SOSF)

From the above considerations, we now propose a new methodology, called System of System Failures (SOSF), to promote double loop learning and satisfy the above three key success factors. Double loop learning is essential for determining whether operating norms (i.e., mental models) are appropriate (Argyris & Schoen, 1996; Morgan, 1986; Senge, 1990). It also provides a meta methodology for changing mental models so as to overcome system improvement shortcomings (Leveson, 2004; Perrow, 1999; Rasmussen, 1997; van Gigch, 1991), as explained in section 2. Among the meta methodologies proposed in a general context, the System of System Methodologies (SOSM) developed by Jackson (Jackson, 2003) is a typical, excellent example. SOSM's main features are the following: i) a meta systemic approach ; i.e. soft system thinking to foster double loop learning (Checkland, 1981; Checkland & Holwell, 1997), and ii) complementarism by encompassing multiple paradigms (contingent approach by combination of various methodologies from various paradigms, depending on problem situations). Figure 2 shows the framework of SOSM. Various classes of systems thinking are located in two-dimensional space, where the two dimensions are participants and systems. The current troubleshooting techniques discussed in section 2 (i.e., FTA, FMEA, IEC) belong to the unitary-simple domain in SOSM.

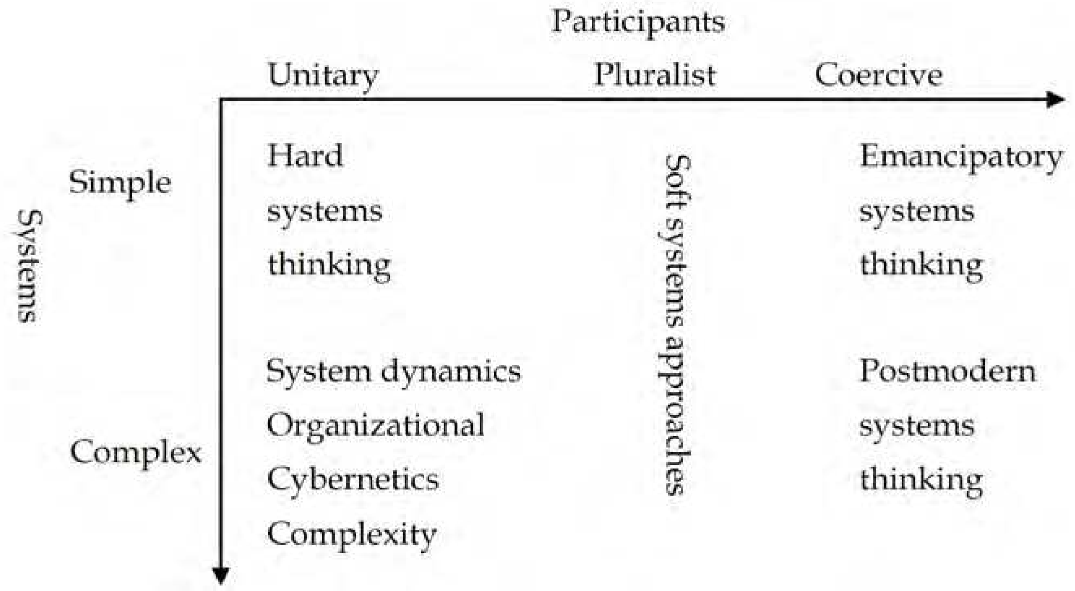


Fig. 2. Systems approaches related to problem context in the System of System Methodologies (SOSM).

In particular, SOSF is designed by allocating each type of failure from a taxonomy of system failures (van Gigch, 1986) into SOSM space (Fig. 3). There is no coercive domain in SOSF, because the main focus of this chapter is technological systems rather than social systems. The stakeholders for achieving engineering safety are covered fully by the unitary and pluralist domains in SOSM. The allocation of each type of failure from SOSM into SOSF is quite straightforward. The structure connecting SOSM and SOSF is shown in Fig. 4. The left-hand side represents layers of abstraction from reality to methodology to meta methodology. In the realm of system failures, a system failure on the bottom line corresponds to the reality layer. The common language (i.e., the taxonomy of failure) corresponds to the methodology layer. A meta failure (i.e., SOSF) corresponds to the meta methodology layer. Therefore, SOSF is an example of SOSM in the realm of system failure. It is worthwhile to mention the recursive feature of SOSF, depending on the viewpoint of the system. If a target system is broken down into subsystems, each subsystem has its own instance of SOSF. Therefore, a technology failure might be a failure of evolution, one level down, from the viewpoint of the subsystem. Furthermore, this failure of evolution might be a failure of regulation, one level higher, from the viewpoint of the system of systems.

To satisfy the third feature (differentiating the three failure classes) pointed out in section 3.1, we should introduce a third dimension, namely, the failure class. Figure 5 expands two-dimensional SOSF (Fig. 3) into three-dimensional SOSF space, with the addition of the system failure dimension.

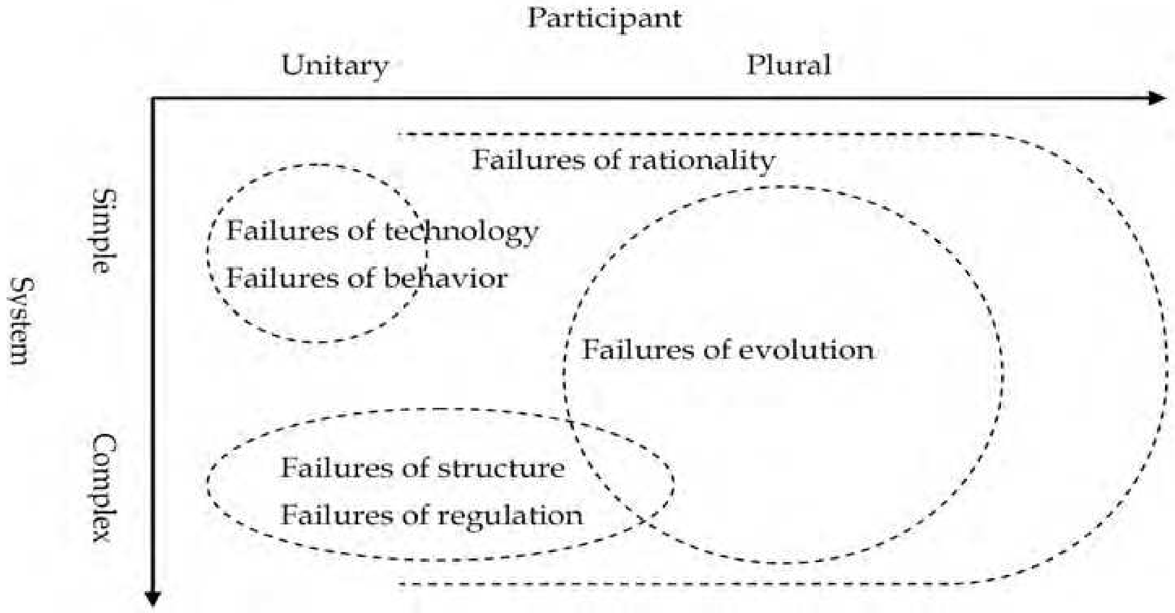


Fig. 3. System of System Failures (SOSF).

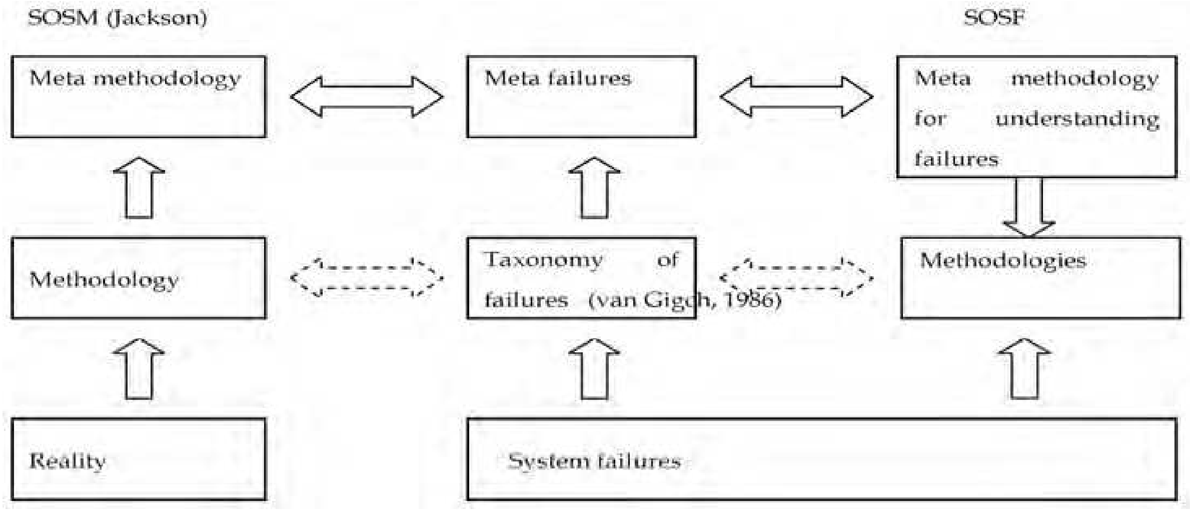


Fig. 4. Meta modeling of system failures and SOSF by using SOSM.

As explained above, because of this recursive nature, it is vital to identify the problem owner in terms of who (i.e., the stakeholder) and where (i.e., the system level in terms of vertical dimension in Table 1).

Table 2 summarizes the general notation of system failures for confirming the mutually exclusive and collectively exhaustive (MECE) nature of the diagnosis, as well as “who,” "where," and "what," which stand for the stakeholder, systems 1 to 5, and the failure class, respectively. The horizontal arrows in Table 2 show that at the same system level, stakeholders should be compared in order to identify responsibilities. If a stakeholder is identified, the system level (1 to 5) and objective (what, who, and when) should be identified using the vertical arrows. This ensures the efficacy of double loop learning by changing the model of the model (i.e., the meta model of the operating norm).

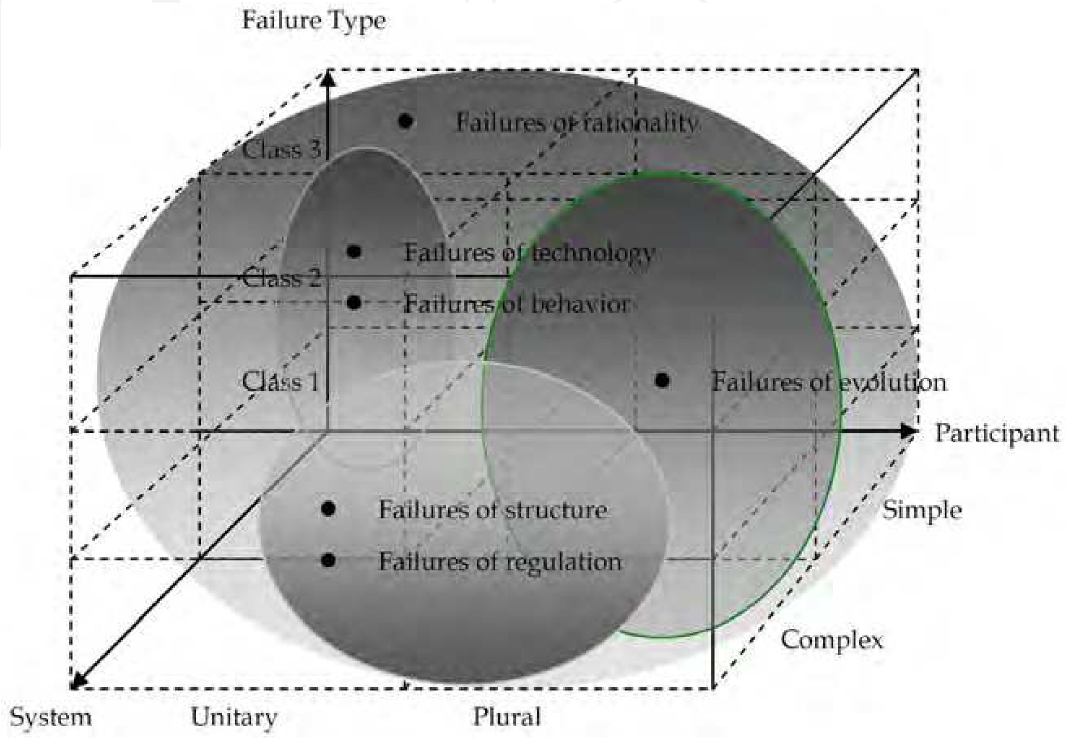


Fig. 5. Three-dimensional SOSF space.

		Stakeholder A	B	C
System 5	Mental model	←		→ Class 3
System 4	Operating norm	←	Class 2	→ ↓
Systems 1-3	Operation	Class 1 ←	↓	→ ↓

Table 2. General notation of system failure.

In the next section, we introduce the two new methodologies that cover the SOSF space.

3.3 Failure factor structuring methodology

We propose new failure factor structuring methodology to overcome system failures caused by complex failure factors (Nakamura & Kijima, 2008a). Generally, complex system failures arise from a variety of factors and combinations of those factors. Since these factors often have a qualitative nature, it is important to have a holistic view that reveals the quantitative relationships among qualitative factors in order to construct an effective methodology. The methodology should address complex system failures in terms of obtaining the observations needed to rectify the worldview of maintenance (i.e., double-loop learning). The failure factor structuring methodology (FFSM) should promote double-loop learning through viewing the system in a holistic way. Figure 6 shows a general overview of this methodology, and Table 3 lists the objectives for each phase of FFSM.

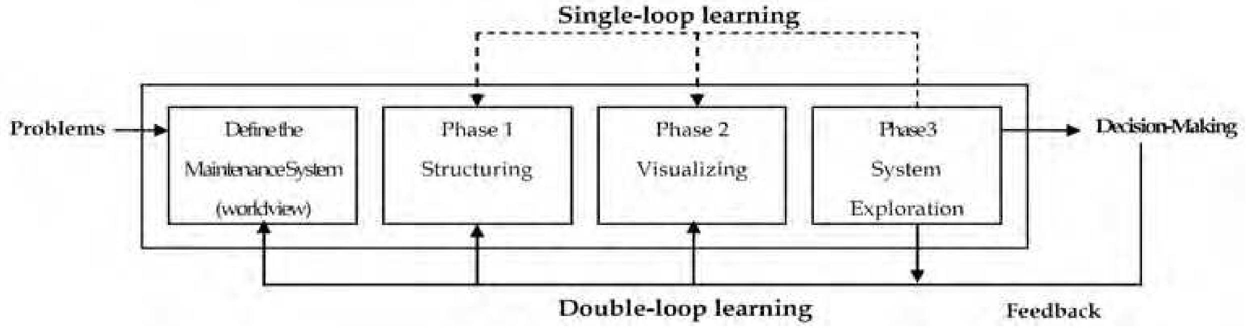


Fig. 6. General overview of FFSM

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

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

3.4 System failure dynamic model

We propose new nonlinear systemic model to overcome system failures caused by environmental changes through time (Nakamura & Kijima, 2008b, 2009a). This “system failure dynamic model (SFDM)” is based on system failure class. 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). 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. Turner and Pidgeon (1997) found that organizations responsible for a failure had “failure of foresight” in common. The failure or the disaster had a long incubation period characterized by a number of discrepant events signaling potential danger. These events were typically overlooked or misinterpreted and accumulated unnoticed. To clarify that mechanism, Turner and Pidgeon decomposed the system lifecycle from the initial development stage to cultural readjustment through catastrophic disasters into six stages (Turner & Pidgeon, 1977, p. 88). They are Stage I: Initial beliefs and norms, Stage II: Incubation period, Stage III: Precipitating event, Stage IV: Onset, Stage V: Rescue and salvage and Stage VI: Full cultural readjustment. The second stage, or incubation period, is hard to identify due to the various side effects of quick fixes (Turner & Pidgeon, 1997). Therefore the second stage is playing the crucial role to lead catastrophic disaster. System failures have specific features corresponding to these six stages. Class 1 failures occur in the early stages, while Class 2 and 3 failures emerge gradually over time. If we have a way to identify the class of a failure, we can prolong the system life cycle by introducing countermeasures. SFDM should be used periodically to ensure that the system behaves as expected (Reason, 1997, 2003) and that side effects due to quick fixes are prevented.

3.5 Relationships among SOSF and related methodologies

The SOSF meta-methodology overcomes the shortcomings of the current methodologies. The current methodologies (i.e., FTA and FMEA) are reviewed through SOSF and the two new methodologies (i.e., FFSM and SFDM) are proposed to complement the shortcoming of the current methodologies. The relationships among SOSF, FFSM, SFDM, and system failures are illustrated in Figure 7.

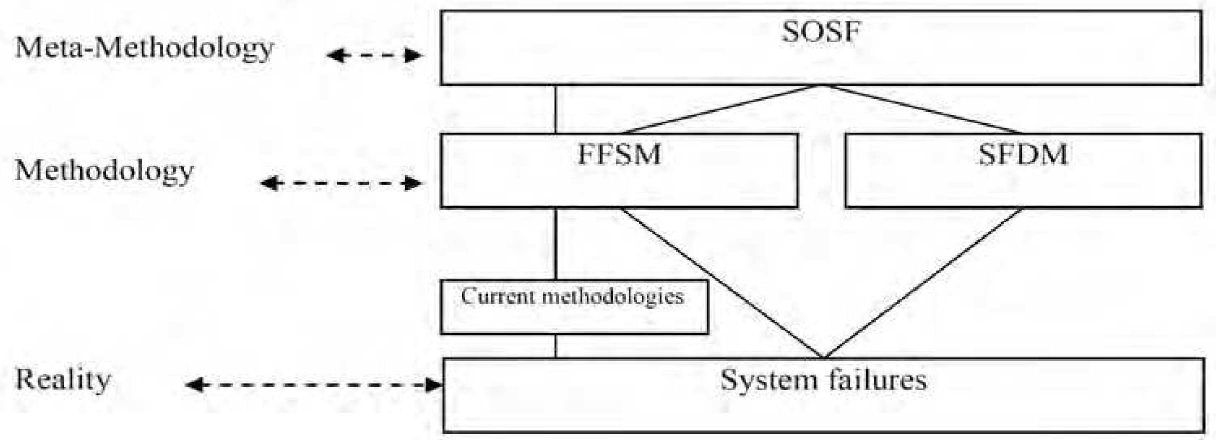


Fig. 7. Relationships among SOSF, FFSM, and SFDM

Table 4 shows the methodology mapping onto SOSF space.

	Within same class	Spread over different classes
Unitary vs. unitary	FTA, FEMA	FFSM
Spread over different domains	SFDM	

Table 4. Methodology mapping to SOSF space

4. Total system intervention for system failure (TSI for SF) methodology as an application procedure

Total system intervention (TSI) is a critical system practice for managing complex and differing viewpoints. In the previous chapter, we introduces meta-methodology called "system of system failures (SOSF)" as a common language among various stakeholders to improve their understanding of system failures. Then we propose the actual application scenario, or "TSI for SF". The SOSF and related methodologies are used in the course of the subsequent discussion and debate to agree upon who is responsible for the failure and to identify the preventative measures to be applied. Flood and Jackson (1991) identified seven principles underpinning the TSI.

First principle: Problem situations are too complicated to understand from one perspective, and the issues they throw up are too complex to tackle with quick fixes.

Second principle: Problem situations, and the concerns, issues, and problems they embody, should therefore be investigated from a variety of perspectives.

Third principle: Once the major issues and problems have been highlighted, a suitable systems methodology or methodologies must be identified to guide intervention.

Fourth principle: The relative strengths and weaknesses of different system methodologies should be appreciated, and this knowledge, together with an understanding of the main issues and concerns, should guide the choice of appropriate methodologies.

Fifth principle: Different perspectives and system methodologies should be used in a complementary way to highlight and address different aspects of organizations and their issues and problems.

Sixth principle: The TSI sets out a systemic cycle of inquiry with interaction back and forth between its three phases.

Seventh principle: Facilitators and participants are engaged at all stages of the TSI process.

Jackson (2006) argues the sixth principle refers to the three phases of the TSI meta-methodology: *creativity, choice, and implementation.* These three phases precede a reflection phase. Therefore, the critical systems practice it embraces is an enhanced version of 'total systems intervention' (Flood & Jackson, 1991), which has four phases: *creativity*, *choice*, *implementation*, and *reflection* (Jackson, 2006).

Based upon the seven principles identified by Flood and Jackson (1991), we introduced new TSI for SF as an application procedure and it has six phases as follows.

4.1 Phase 1. Become aware of system failure relating to the first principle

Owners of issues and problems understand that they are too complicated to understand from one perspective, and the issues they throw up are too complex to tackle with quick fixes.

4.2 Phase 2. Identify stakeholders relating to the second principle

Owners of issues and problems should identify stakeholders relating to the issues or problems from phase 1.

4.3 Phase 3. Creativity: Identify metaphors relating to the third and the creativity phase in the sixth principle

In the creativity phase, the many different possible views of organizations and their problems should be recognized, and managers and analysts should be encouraged to explore them through the use of Morgan's (1986) "images or metaphors," particularly the machine, organism, brain, culture, and coercive system metaphors. The aim is to take the broadest possible critical look at the problem situation but gradually to focus on those aspects currently most crucial to the organization (Jackson, 2006).

In order to understand system failures, we need models and metaphors. Then methodologies are developed depending upon those metaphors. We introduce three system failure models with metaphors (i.e., the third principle).

4.3.1 Simple linear system failure model (Domino metaphor)

The archetype of a simple linear model explains system failure as the linear propagation of a chain of causes and effects (Heinrich et al., 1980). Figure 8 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.

Fig. 8. Domino metaphor

4.3.2 Complex linear system failure model (Swiss cheese metaphor)

The archetype of a complex linear model is well known Swiss cheese model (Fig. 9) first proposed by Reason (1997, 2003). The model put the importance on latent as well as manifested causes. The authors proposed FFSM (Nakamura & Kijima, 2008a, 2009b) as surfacing hidden (latent) factors to suppress deviations leading to system failures.

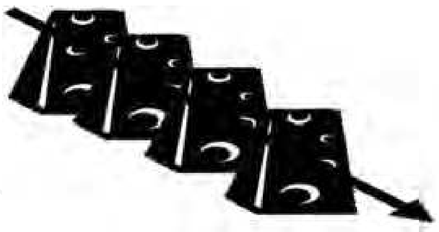


Fig. 9. Swiss cheese metaphor

4.3.3 Non linear or systemic model (Unrocking boat metaphor)

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 (1997, 2003) explains the organizational life span between protection and catastrophe. The lifespan of a hypothetical organization through production-protection space (Fig. 10) explains why organizational accidents repeat, with this history ending in catastrophe. This is why the periodic application of the methodology prolong system life cycle.

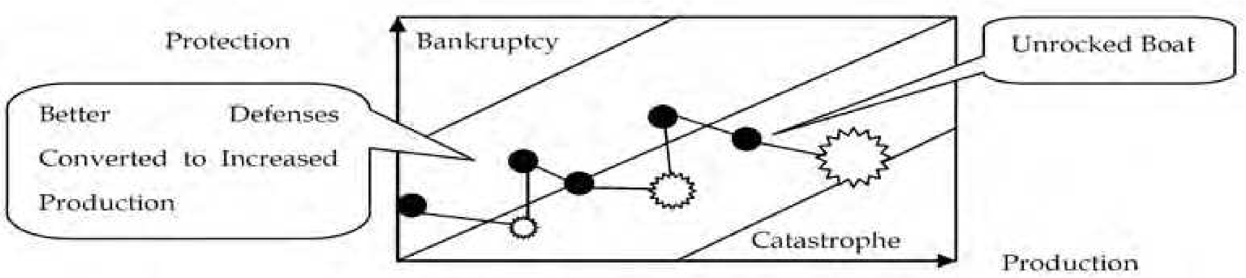
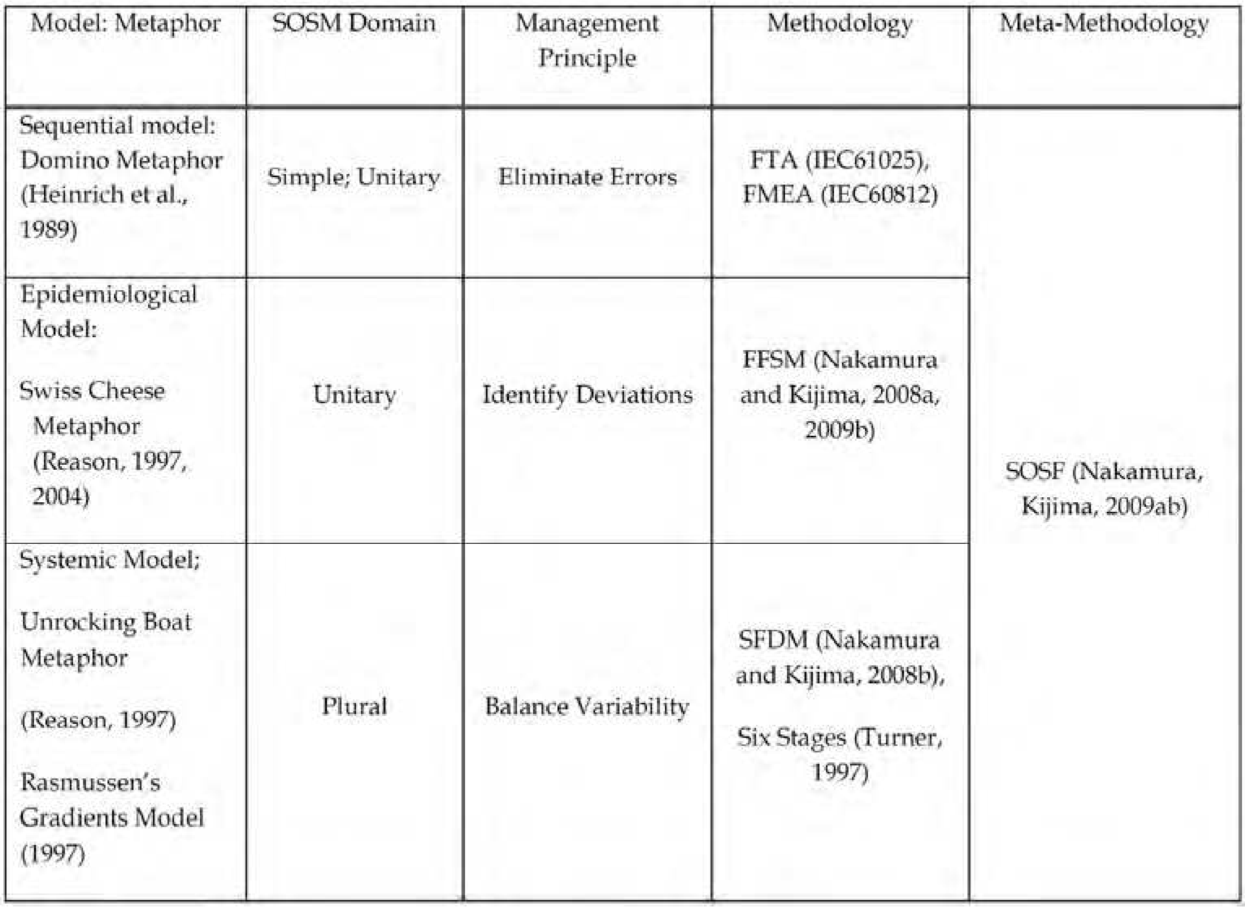


Fig. 10. Lifespan of a hypothetical organization through production-protection space

4.4 Phase 4. Choice: Select methodology using SOSF meta-methodology relating to the fourth and the choice phase in the sixth principle

In this phase, the metaphors generated in the creativity phase are mapped to the SOSF space (Nakamura & Kijima, 2009a) to match the methodology to the problem situation. In the SOSF meta-methodology, problem situations are mapped using three axes (simple/complex, unitary/plural, and Class 1/2/3) in accordance with the degree of (dis)agreement between participants. Problem situations are then mapped to the methodologies as outlined in Table 5. Note that the SOSF meta-methodology is used not to deterministically prescribe which methodology to choose but to illuminate and inform that choice (i.e., the fourth principle).



Model: Metaphor	SOSM Domain	Management Principle	Methodology	Meta-Methodology
Sequential model: Domino Metaphor (Heinrich et al., 1989)	Simple; Unitary	Eliminate Errors	FTA (IEC61025), FMEA (IEC60812)	SOSF (Nakamura, Kijima, 2009ab)
Epidemiological Model: Swiss Cheese Metaphor (Reason, 1997, 2004)	Unitary	Identify Deviations	FFSM (Nakamura and Kijima, 2008a, 2009b)	
Systemic Model; Unrocking Boat Metaphor (Reason, 1997) Rasmussen's Gradients Model (1997)	Plural	Balance Variability	SFDM (Nakamura and Kijima, 2008b), Six Stages (Turner, 1997)	

Table 5. Three system-failure models and their approach to management

We introduce a matrix that clarifies the differences in opinion among stakeholders. Using it helps to clarify the stakeholder views and to identify stakeholders with opposing views. In the example stakeholder matrix in Fig. 11, stakeholders “a” and “b” have opposing views, as shown on the left. After they discuss and debate their views, stakeholder “a” takes responsibility, as shown on the right. In short, a diagonal matrix is created from a non-diagonal one. Table 5 summarizes the system failure models and related methodologies as well as the meta-methodology.

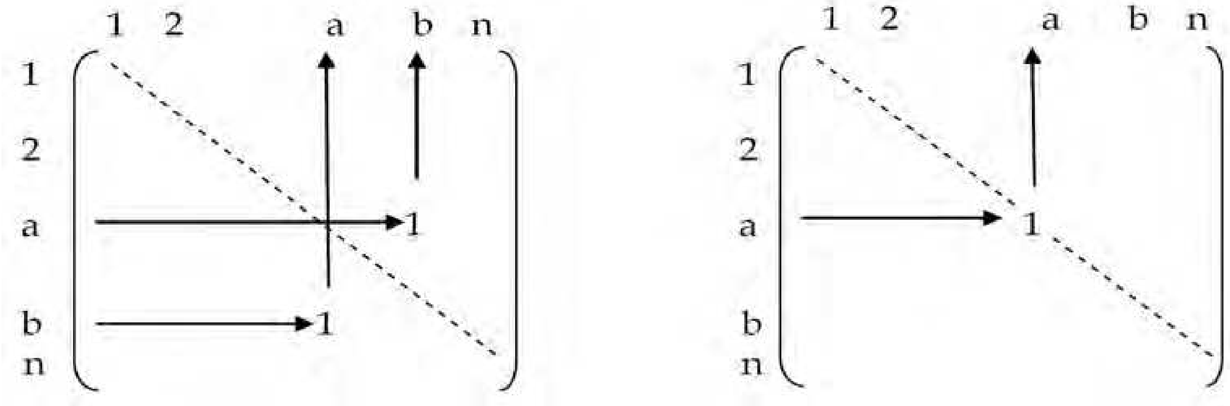


Fig. 11. Stakeholder matrix

4.5 Phase 5. Implementation: Take action relating to fifth and the implementation phase in the sixth principle

In the implementation phase, methodologies are applied to produce change. The methodologies should be used in a complementary way to highlight and address different aspects of organizations and their issues and problems (i.e., the fifth principle). In this phase, the selected methodology in table 5 could be used in accordance with the complementary principles of TSI.

4.6 Phase 6. Reflection: Acquire new learning relating to the reflection phase in the sixth principle

In the reflection phase, the intervention should be evaluated and learning about the problem situation, the meta-methodology itself, the generic system methodologies, and the specific methods used should be produced. The outcome is research findings that are used, for example, as feedback for improving earlier stages of the meta-methodology (i.e., Fig. 12). The relationship between the stages is shown in Fig. 12. There are two feedback loops in Fig. 12. One is to the metaphors (phase3) and the other is to the methodologies (phase4).

5. Application to ICT systems

This section discusses an example application of the TSI for SF methodology to an ICT system failure caused by an operator error resulting from a misunderstanding of the product specifications. In this case, the operator or users who use the products in question was responsible for the failure. The incident escalation procedure is shown in Fig. 13. Those users who encounter the problems of the products report the incident to the help desk, and the help desk provides them with a solution. The help desk then identifies the cause of the incident, and, if it was caused by faulty product design, the help desk escalates it to the development section for further investigation. The development section designs new products on the basis of data for the escalated incidents that the help desk believes were due to product defects. This is mainly because the user-related incidents are screened at the help desk so that the development section can concentrate on product-related issues. The development section measures product quality by AFR (Annual Failure Rate) using only the incidents escalated from the help desk, not by ACR (Annual Call Rate) using all the incidents received directly from the users. AFR is introduced to measure a product quality not to measure a system quality. Therefore AFR is a part of ACR. The metric for product quality is the AFR and system quality that includes product quality is the ACR, which are calculated as shown in Fig. 14.

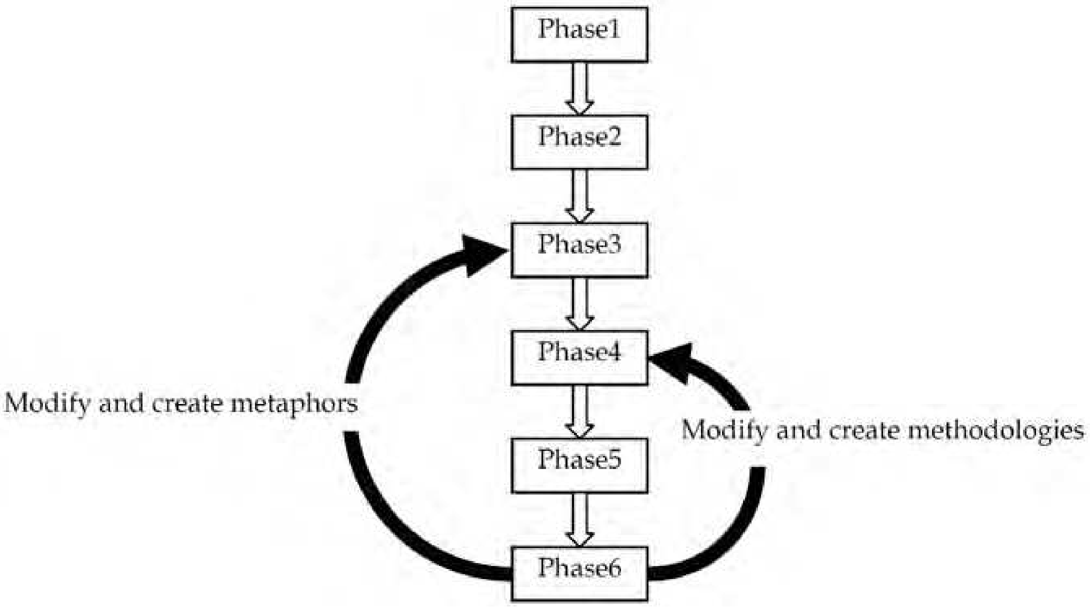


Fig. 12. The application flow for TSI for SF

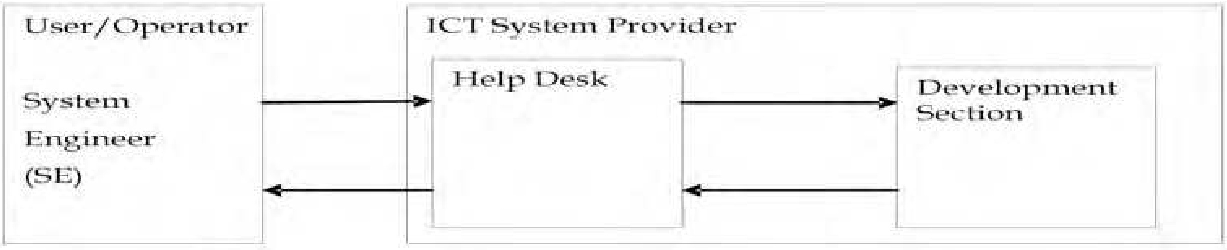
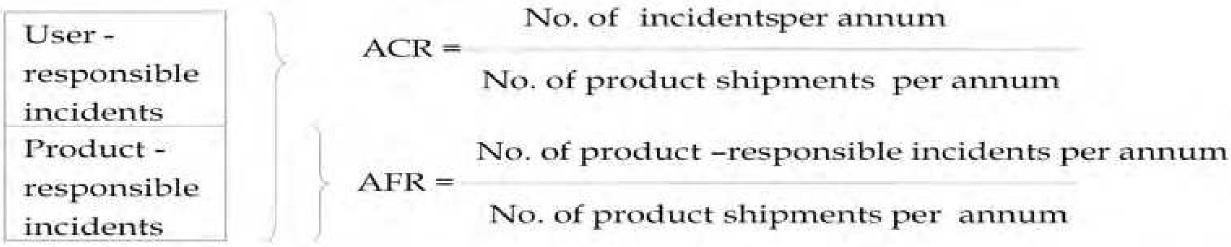


Fig. 13. Incident escalation procedure



$$\mathrm{ACR} = \frac{\text{No. of incidentsper annum}}{\text{No. of product shipments per annum}}$$

$$\mathrm{AFR} = \frac{\text{No. of product –responsible incidents per annum}}{\text{No. of product shipments per annum}}$$

Fig. 14. Calculation of annual failure rate (AFR) and annual call rate (ACR)

As mentioned above, there are six phases in the application procedure for TSI for SF. The followings are the summary of the actual application example.

5.1 Phase 1. Become aware of system failure

In the first stage of intervention, the development section believes that the quality of their product is superior to the average quality of its competitors' products on the basis of internal benchmarking. A third party customer survey reveals that customers judge the quality to be less than that revealed by the internal benchmarking. Upon learning of this discrepancy, the system quality assurance (SQA) section of the ICT system provider sets up a working group to identify the problems.

5.2 Phase 2. Identify stakeholders

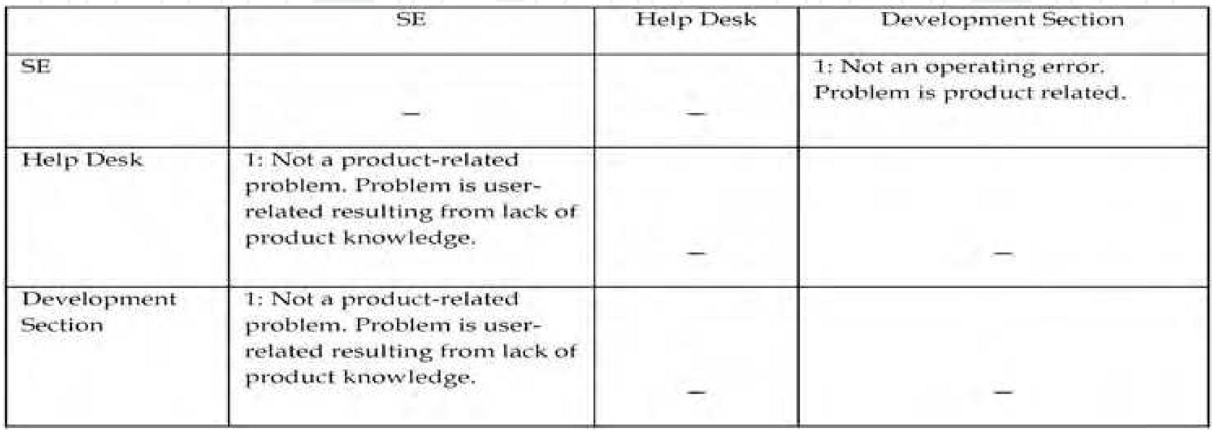
The owner of the working group, the SQA section, identifies three stakeholders: an SE (representing a user or operator), the help desk representing the first line engineer, and the development section representing the second line engineer.

5.3 Phase 3. Creativity: Identify metaphors

The SQA section identifies the difference in the key performance indicators (KPIs) between the help desk and the development section. The help desk's KPIs are mainly related to the processing speed and the development section is to the AFR. The SQA section recognizes that increasing the speed should not increase the number of incidents escalated to development section. Furthermore, one way to improve the AFR is to close incidents as user responsible incidents (Fig. 13). Obviously, this may not the best way to handle incidents. Therefore, the two sections' KPIs are not user oriented. The SQA section identifies the unrocking boat metaphor (Table 5) as appropriate for this situation (i.e., the organization is drifting through the environment between excessive economic gain and safety).

5.4 Phase 4. Choice: Select methodology using SOSF meta-methodology

The stakeholder opinions are clarified using the stakeholder matrix (Fig. 11) in order to identify stakeholders with opposing views. As shown in Table 6, the SE and development section have opposing views. The Help desk claims that the SE made an error in operation.



	SE	Help Desk	Development Section
SE	—	—	1: Not an operating error. Problem is product related.
Help Desk	1: Not a product-related problem. Problem is user-related resulting from lack of product knowledge.	—	—
Development Section	1: Not a product-related problem. Problem is user-related resulting from lack of product knowledge.	—	—

Table 6. Stakeholder matrix

The SQA section uses the SFDM to identify three archetypes:

- misunderstanding a Class 2 or 3 failure as a Class 1 failure, (problem)
- erosion of safety goals accompanied by incentive to report fewer incidents (side effect), and
- fix that fails (side effect).

5.4.1 Misunderstanding Class 2 or 3 failure as Class 1 failure (problem)

The source of the failure is inside the help desk system boundary (i.e., a Class 1 failure) although the actual cause is outside the boundary. This archetype (Fig. 15) explains why system failures reoccur following a quick fix or an inappropriate fix. Such fixes might reduce the number of system failures in the short term, but the effects of such fixes gradually become saturated at a level below the organization's goal (i.e., target) level. The balancing intended consequence (BIC) loop becomes open, so quick fixes have no further effect. The balancing unintended consequence (BUC) loop also becomes open as a result of misunderstanding the system failure class and not introducing an effective solution. The sequence of this archetype is from (1) to (5) in Fig. 15. Arrow (1) with the "+" sign indicates that an increase in the number of Class 1 failures causes an increase in the number of actions. Arrow (2) with the "+" sign indicates that the increase in the number of actions increases the number of quick fixes. Arrow (3) indicates that the increase in the number of quick fixes contributes slightly to reducing the number of Class 1 failures. The root cause is outside the system boundary and is unaffected by arrow (4). Therefore, arrow (5) with the "+" sign indicates that the root cause increases the number of Class 1 failures.

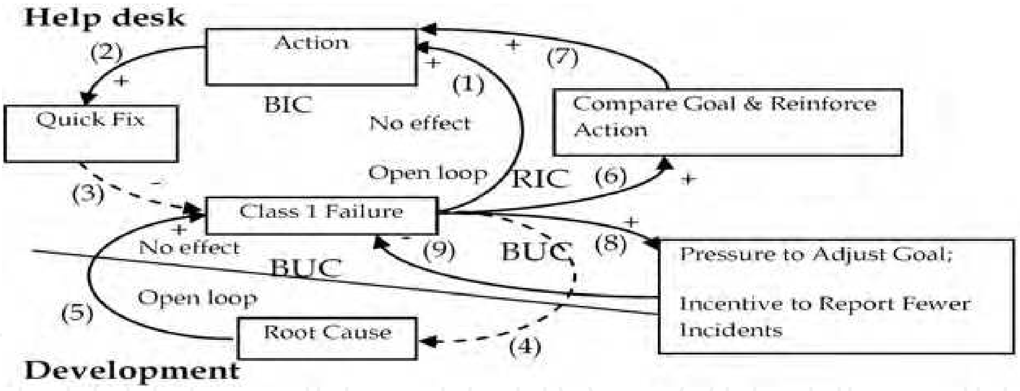


Fig. 15. Misunderstanding system failure archetype

The archetype shown in Fig. 15 is a single-loop learning scenario—a reinforcing action is introduced that is based on the deviation from a predetermined goal. The reinforcing intended consequences (RICs) action to improve the situation leads to the introduction of additional quick fixes, which simply leads to the repetition of a similar scenario. The sequence of this archetype is from (6) to (7) in Fig. 15. Arrow (6) with the "+" sign indicates that an increase in the number of Class 1 failures reinforces the compare goal and reinforce action. Arrow (7) with the "+" sign indicates that reinforcing the compare goal and adjust action increases the number of actions. The RICs action causes various side effects, including erosion of safety goals accompanied by an incentive to report fewer incidents. These side effects are hard to detect because the performance malfunction alarm is muted, and management can identify these effects only by quantitatively measuring performance. This explains why a single-loop learning solution for improving system performance is bound to fail, as Van Gigch (1991) pointed out. In this situation, the root cause outside the system boundary must also be addressed.

5.4.2 Erosion of safety goals accompanied by incentive to report fewer incidents

This side effect is introduced when the RICs loop becomes tighter without a further reduction in the number of system failures (Fig. 15). Increased pressure to achieve the goal emerges from the BUC loop in the form of shifting the goal (i.e., lowering it) and/or hiding the actual state of quality or safety from management. In this relative achievement scenario, a manager who stays within the system boundary has difficulty detecting the actual state of achievement. This is why many Japanese manufacturers have the slogan "3R-ism," which reminds managers to identify a problem at a "real site," confirm it with "real objects," and discuss it with a "real person in charge" before taking any action. The sequence of this archetype is from (8) to (9) in Fig. 15. Arrow (8) with the "+" sign indicates that an increase in the number of Class 1 failures causes pressure to adjust the goal or creates an incentive to report fewer incidents. Arrow (9) with the "−" sign indicates an increase in the number of Class 1 failures that are hidden.

5.4.3 Fix that fails archetype (side effect)

The source of the failure is outside the help desk's system boundary. Figure 16 illustrates a typical example of local optimization. The action taken for the root cause is a short-term solution to the problem that introduces delayed, unintended consequences outside the system boundary, resulting in a Class 2 or 3 failure. 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 reinforcing unintended consequence (RUC) loop to further increase the number of system failures. This out-of-control situation can only be managed at the expense of others and damages the organization in the long term. The sequence of this archetype is from (1) to (6) in Fig. 16. Arrow (1) with the "+" sign indicates that an increase in the number of Class 2 or 3 failures increases the number of actions within the system boundary. These actions do not attack the root cause (i.e., dotted arrow (5)). Therefore, arrow (2) with the "+" sign has no effect on reducing the number of Class 2 or 3 failures. Alternatively, the arrow with the time-delay symbol (=) might increase the number of Class 2 or 3 failures because of local optimization side effects. Arrows (3) and (4) with the "+" sign introduce an adjust goal and reinforce action without further reducing the number of Class 2 or 3 failures. Arrows (5) and (6) are not in effect during this phase of the archetype.

In this application example, as a result the stakeholders reached the broader and holistic understanding using the SOSF meta-methodology. At initial stage (i.e., preceding stage 5), the user thought these errors are not operation-related but product-related. Conversely, the development section thought they are operation-related. Therefore, the user insisted that they are Class 3 failures of evolution in complex and plural domains in SOSF. Conversely, the development section insisted that they are Class 1 failures of behavior in a simple and unitary domain. Figure 17 illustrates the SOSF space showing all stakeholder opinions.

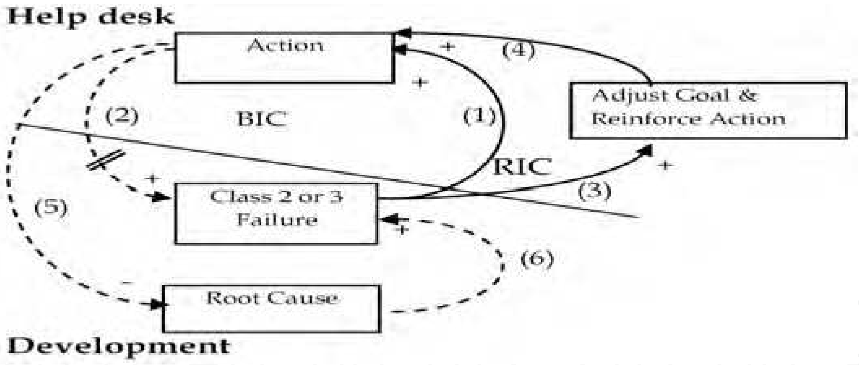


Fig. 16. Fix that fails archetype (side effect)

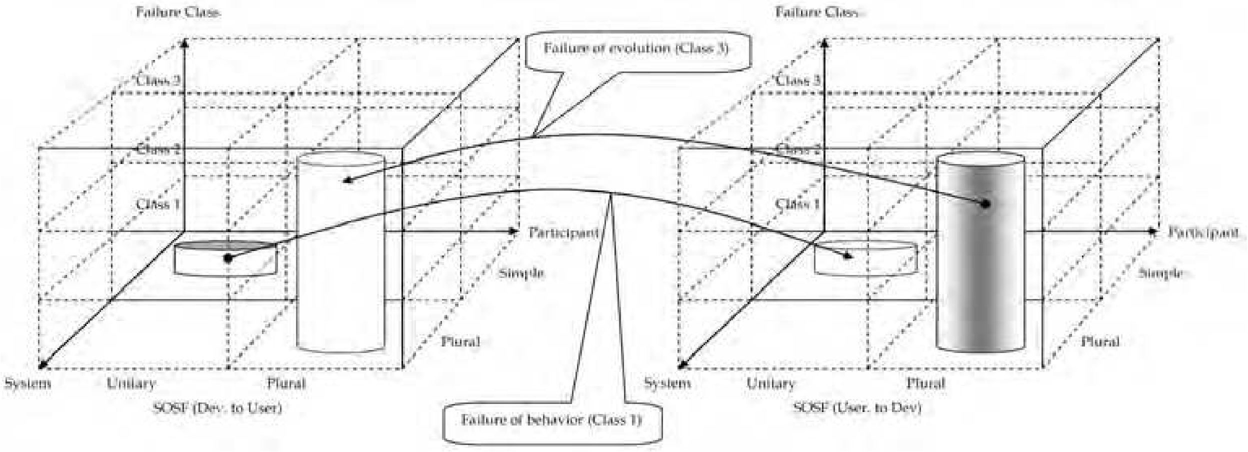
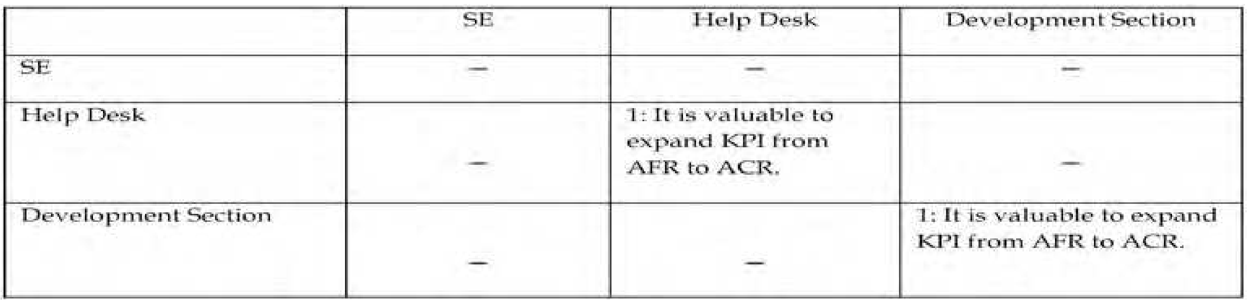


Fig. 17. Simple-Unitary (Class 1) vs. Complex-Plural (Class 3)

5.5 Phase 5. Implementation: Take action

After the debate and discussion, the stakeholders reached the conclusion shown in Table 7.



	SE	Help Desk	Development Section
SE	–	–	–
Help Desk	–	1: It is valuable to expand KPI from AFR to ACR.	–
Development Section	–	–	1: It is valuable to expand KPI from AFR to ACR.

Table 7. Clarify stakeholder opinions using matrix

The SQA section analyzed the user-related incidents and, as illustrated in Fig. 18, judged that 36% of them were possibly product-related. Following their debate and discussion, the SQA section, the help desk, and the development section agreed to change their KPI from the AFR to ACR. The incident reduction scheme is illustrated in Fig. 18. Over the two years of the operation with the new KPI, the ACR have been reduced respectively by approximately 52, 17, 51, and 19% for products A, B, C, and D with the overall average of 36% reduction in Fig. 19.

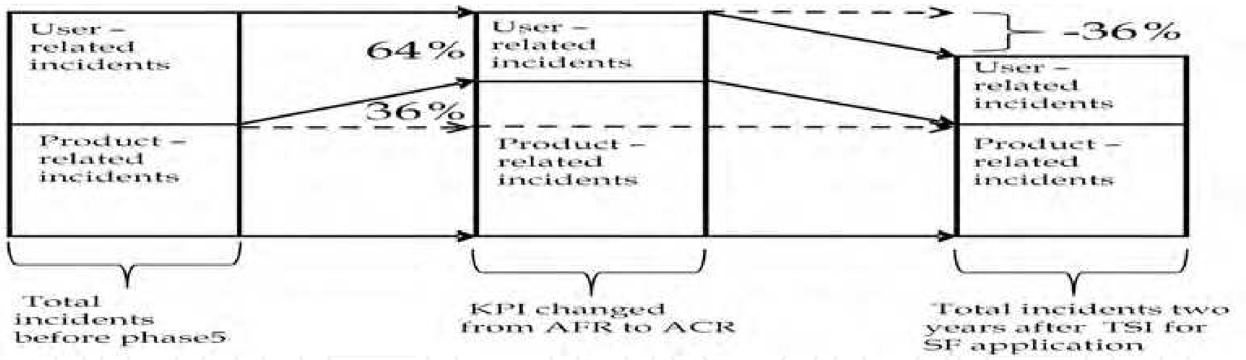

User – related incidents
Product – related incidents
64%
36%
User – related incidents
Product – related incidents
-36%
User – related incidents
Product – related incidents
Total incidents before phase5
KPI changed from AFR to ACR
Total incidents two years after TSI for SF application

Fig. 18. Incidents transition over two-year period

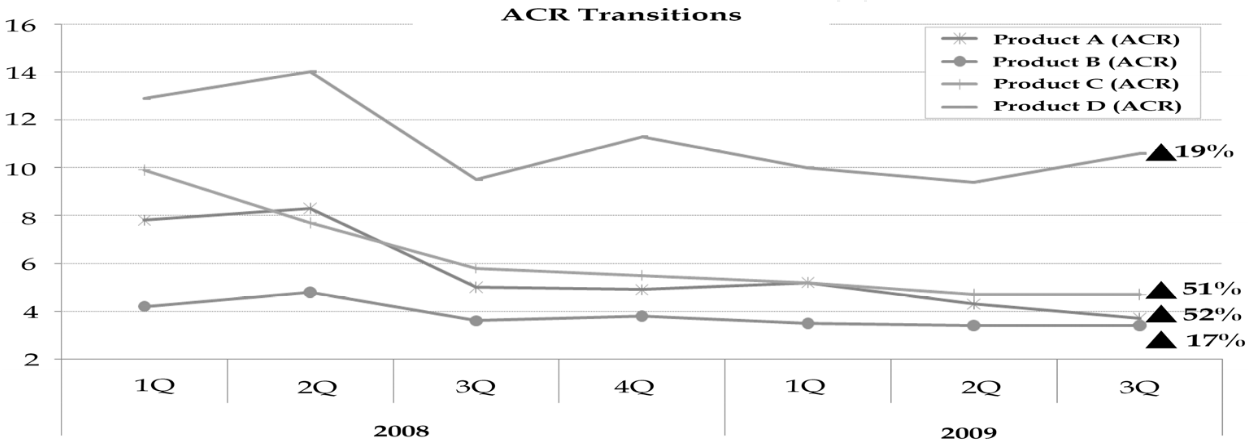


Fig. 19. ACR transitions

5.6 Phase 6. Reflection: Acquire new learning

On the basis of the application example described above, we can identify three ways to overcome the problem of misunderstanding a Class 2 or 3 failure as a Class 1 failure: introduce an absolute goal, close the gap between stakeholders, and enlarge the system boundary. All three actions promote double-loop learning because they alter the process design to improve system quality or safety. In contrast, single-loop learning leads to side effects, as explained for phase four:

- erosion of safety goals and creation of incentive to report fewer incidents, and
- failure of a previous fix.

There are three double-loop learning archetypes.

5.6.1 Double-loop learning for Class 2 failure archetype (solution)

As noted above, it is necessary to focus on the possibilities of relative achievement or the side effects of a quick fix. A tacit assumption of a gap between stakeholders should be surfaced throughout the discussion and debate to close the responsibility gap. Application of this solution to the scenario shown in Fig. 15, misunderstanding system failure archetype, is illustrated in Fig. 20. The sequence of this archetype is from (1) to (6). Arrow (1) with the "+" sign indicates that an increase in the number of Class 2 failures increases the number of actions within the system boundary. These actions induce various side effects (erosion of safety goals or reporting fewer incidents), as discussed above. Arrow (2) with the "+" sign indicates reviewing the stakeholders' mental model gap and redefining or adjusting the ultimate goal. Arrow (3) with the "+" sign indicates provoking a new action. Arrow (4) with the "−" sign indicates that the new action attacks the root cause, which resides outside the system boundary. Arrow (5) with the "+" sign indicates eventually reducing the number of Class 2 failures. Arrow (6) with the "+" sign indicates the path to adjusting the goal and defining the ultimate solution.

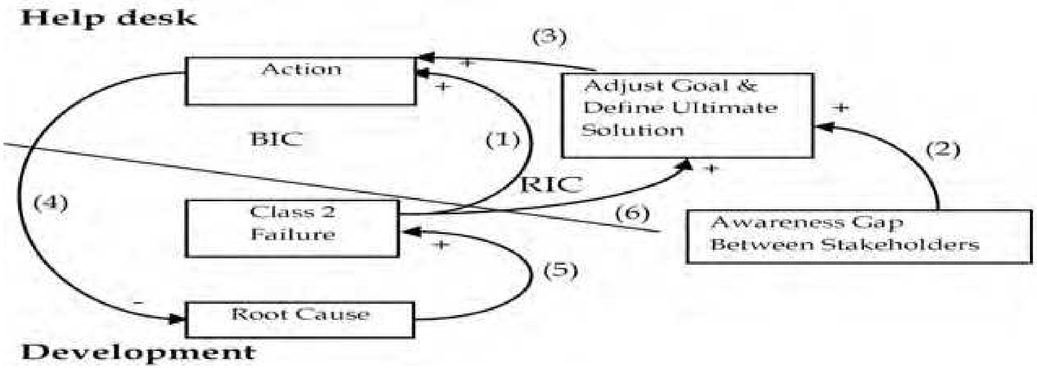


Fig. 20. Double-loop learning for Class 2 failure (solution)

5.6.2 Double-loop learning for class 3 failure archetype (solution)

As mentioned in the introduction, the speed of technology advancement and the growth of complexity are unpredictable. Therefore, a current goal could later become obsolete. This could be the root cause of a 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. The sequence of this archetype is from (1) to (8) in Fig. 21. Arrow (1) with the "+" sign indicates that an increase in the number of Class 3 failures increases the number of actions within the system boundary. These actions do not attack the root cause, so there is no effect on reducing the number of Class 3 failures, as indicated by arrow (2). Arrows (3) and (4) with "+" signs indicate introducing the ideal goal, provoking awareness of the gap between the current and ideal Goals, and adjusting the goal and defining the ultimate solution. Arrow (5) with the "+" sign indicates introducing a new action, and arrow (6) with the "−" sign indicates attacking the root cause, which reduces the number of Class 3 failures, as arrow (7) indicates. Arrow (8) with the "+" sign indicates further enhancement of adjust goal and define ultimate solution.

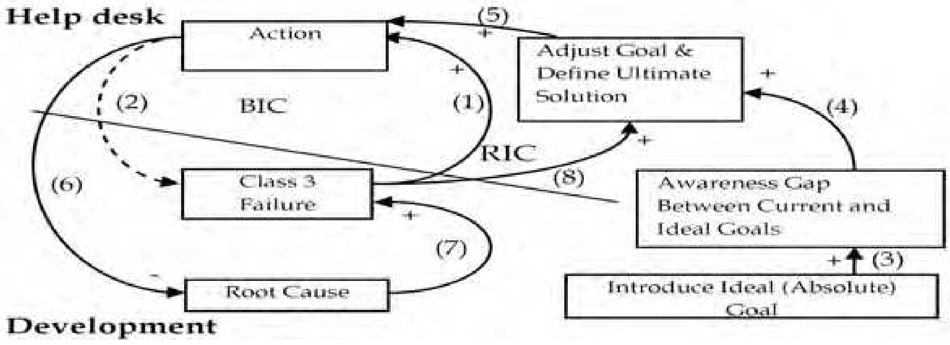

Help desk
Action
(5)
+
Adjust Goal & Define Ultimate Solution
+
+
(2)
BIC
(1)
(4)
RIC
+
(6)
(8)
Class 3 Failure
Awareness Gap Between Current and Ideal Goals
+
(7)
+
(3)
-
Root Cause
Introduce Ideal (Absolute) Goal
Development

Fig. 21. Double-loop learning for Class 3 failure (solution).

5.6.3 Double-loop learning for fix that fails archetype (solution)

The solution for this archetype is to raise the viewpoint of the problem (Fig. 22). Class 2 and 3 failures become Class 1 if the presumed system boundary is enlarged. The sequence of this archetype is from (5) to (7) in Fig. 22. Arrow (5) indicates enlarging the system boundary to incorporate the root cause. This converts Class 2 and 3 failures into Class 1 failures. Arrow (6) with the "−" sign indicates attacking of the root cause, which reduces the number of Class 2 or 3 failures, as indicated by arrow (7).

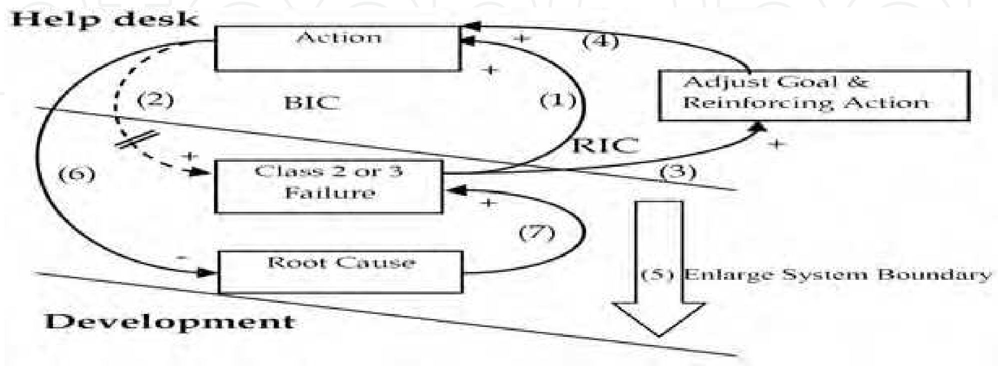


Fig. 22. Double-loop learning for fix that fails archetype (solution)

Figure 23 summarizes the result of SFDM from problem archetype to solution archetype. It shows introducing quick fix (reinforcing current action) is only causing various effects (Erosion of safety goals; incentive for reporting fewer incidents and Fix that fails archetypes).

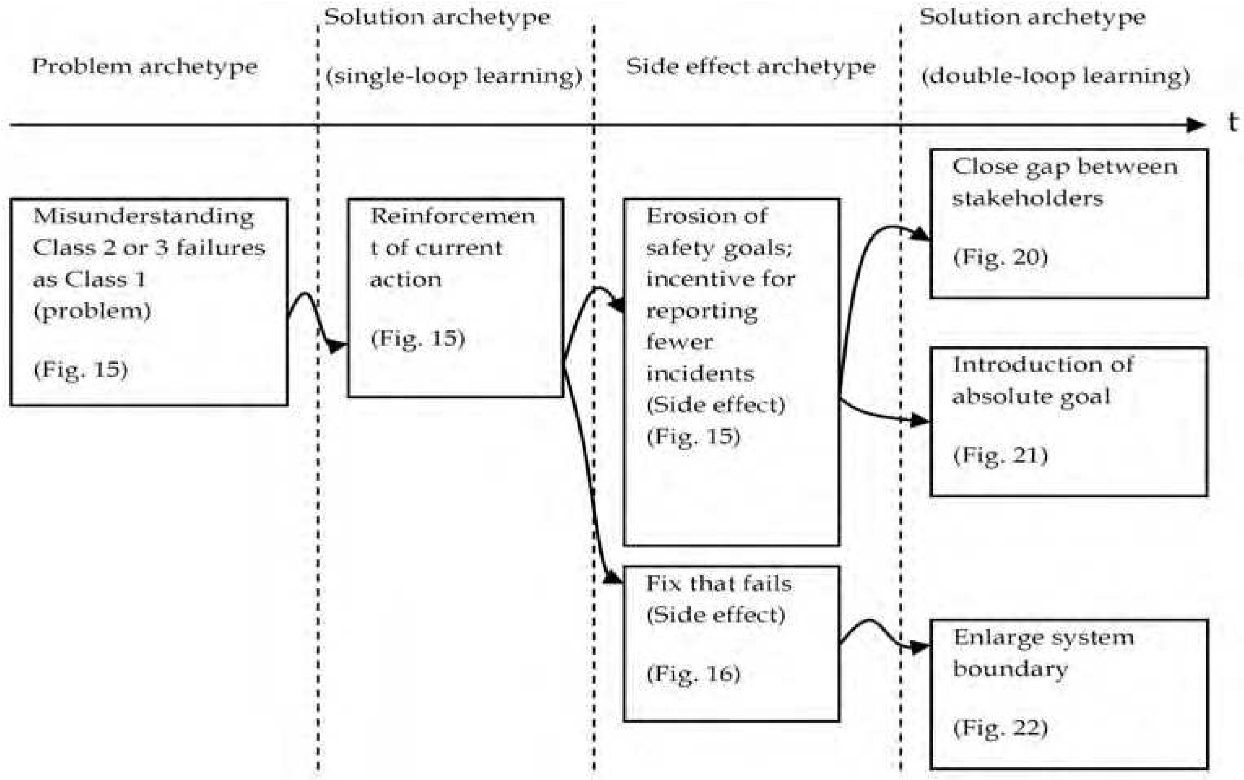

Problem archetype
Solution archetype
(single-loop learning)
Side effect archetype
Solution archetype
(double-loop learning)
t
Misunderstanding Class 2 or 3 failures as Class 1 (problem) (Fig. 15)
Reinforcemen t of current action (Fig. 15)
Erosion of safety goals; incentive for reporting fewer incidents (Side effect) (Fig. 15)
Fix that fails (Side effect) (Fig. 16)
Close gap between stakeholders (Fig. 20)
Introduction of absolute goal (Fig. 21)
Enlarge system boundary (Fig. 22)

Fig. 23. Problem and solution archetypes in engineering system failures through time.

6. Conclusion

In the ICT engineering arena, the predominant methodologies for promoting system quality and safety are deeply rooted in hard systems thinking. Most organizational processes are reductionist approach. This is reasonable to some extent. Engineers in the development section see systems as the combination of components. The quality of these components determines the quality of the system if the system boundary is defined within the aggregation of components. Therefore, the key performance indicators they use for daily routine processes are not drawn from outside the defined system. In the hard systems thinking paradigm, an efficient approach is to identify deviances from the internal goals and rectify them. The predominant techniques and methodologies play a major role in the simple unitary domain of the meta-methodology called "system of system failures (SOSF)". However in a complex and pluralistic stakeholder's environment, it is clear that several side effects were detected in the "system failure dynamic model (SFDM)" process. This is mainly because the discussion and debate is done among different system levels of stakeholders. The third SOSF dimension represents the responsible system class in VSM terminology. The debate between system 1 and system 5 from different stakeholders could introduce unwanted side effects, as explained in section 5. Especially in the case of failure of evolution in pluralistic contexts, representatives of opposing stakeholders should be from system 5. It is particularly effective in critical system practice, even in the ICT engineering arena, to expand the focus to not only 'work; technical interest' but to 'interaction; practical interest'. The "total system intervention for system failure (TSI for SF)" methodology is useful for changing to an absolute goal learning from the gap between stakeholders and enlarging the system boundary.

We conclude with a summary of the checkpoints and corresponding actions.

Checkpoint 1: Is there a recognizable gap between the perceptions of the stakeholders? If not, use the stakeholder matrix to clarify them.

Action1: Close the gap between the stakeholders. The debate should be conducted with the same system level from stakeholders.

Checkpoint 2: Is your KPI related to absolute goal? (i.e., absolute customers) Do your customers know your KPI? If not, assess the viability of introducing absolute goals.

Action2: Introduce absolute goals to avoid local optimization and to ensure that the essential goal is pursued.

Checkpoint 3: Is the system boundary clear? If not, clarify the boundary. If yes, discuss the feasibility and effectiveness of enlarging the boundary.

Action3: Enlarge system boundary. This would enable to reexamine current system boundary and effectiveness of the process. This could be useful to find out side effects.

7. References

Argyris, C. & Schoen, D. (1996). *Organizational Learning II*, Addison Wesley, 0201629836, Mass.

Beer, S. (1979). *The Heart of Enterprise,* John Wiley & Sons, 0471275999, London and New York

Beer, S. (1981). *Brain of the Firm, 2nd edition*, John Wiley & Sons, 0471276871, London and New York

Checkland, P. (1981). *System thinking, system practice*, John Wiley & Sons, 0471279110, UK

Checkland, P. & Holwell, S. (1997). *Information, Systems and Information Systems making sense of the field,* John Wiley & Sons, 0471958204,UK

Flood, R.L. & Jackson, M.C. (1991). *Creative Problem Solving: Total Systems Intervention*, Wiley, 9780471930525, Chichester

Heinrich, H.W.; Petersen, D. & Roos, N. (1980). *Industrial Accident Prevention: A Safety Management Approach. 5th ed,* McGraw-Hill, 0070280614, New York

Jackson, M. C. (2003). *System Thinking - Creative holism for Managers*, John Wiley &Sons, 0470845228, UK

Jackson, M. C. (2006). Creative Holism: A Critical Systems Approach to Complex Problem Situations, *Systems Research and Behavioral Science Vol. 23, Issue 5*, (September/October 2006), pp(647-657)

IEC homepage, 30.04.2011, available from http://www.iec.ch/

IEC 60812 (2006), *Procedure for failure mode and effect analysis (FMEA)*, 4.05.2011, available from http://webstore.iec.ch/webstore/webstore.nsf/artnum/035494/

IEC 61025 (2006), *Fault tree analysis (FTA),* 4.05.2011, available from *http*://webstore.iec.ch/webstore/webstore.nsf/artnum/037347/

ISO homepage, 30.04.2011, available from http://www.iso.org/iso/home.htm/

Kickert, W. J. M. (1980). *Organization of decision-making,* North Holland, 0444854290, Amsterdam

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

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

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

Nakamura, T. & 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, pp(29–47)

Nakamura, T. & Kijima, K. (2009b). A methodology to prolong system lifespan and its application to IT systems. *Proceeding of the 53rd Annual Meeting of the ISSS,* Brisbane, Jul. 2009

Morgan, G. (1986). *Images of Organization,* Sage Publications, 0803928300, California.

Perrow, C. (1999). *Normal Accidents Living with High-Risk Technologies,* Princeton Paperbacks, 0691004129, New York

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

Reason, J. (1997). *Managing the risk of organizational accidents*, Ashgate Publishing limited, pp(3-5), 1840141042, UK

Reason, J. & Hobbs, A. (2003). *Managing Maintenance Error: A Practical Guide*, Ashgate Pub Ltd, 9780754615910, UK

Senge, P. (1990). *The Fifth Discipline: The Art and Practice of the Learning Organization, 1st edition,* Doubleday, 9780385260947 New York

Turner, B. A. & Pidgeon, N. F. (1997). *Man-Made Disasters 2nd edition. Butterworth-Heinemann*, 0750620870, UK

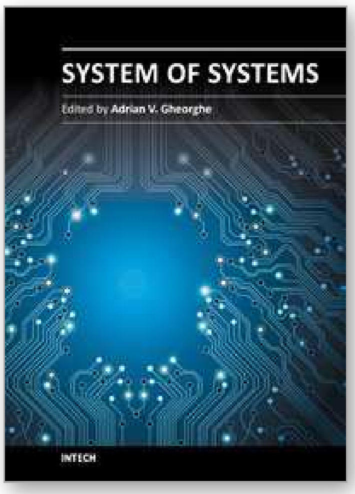
The Columbia Accident Investigation Board Report, 30.04.2011 available from http://history.nasa.gov/columbia/CAIB_reportindex.html, chapter 6 pp(130), chapter 8 pp(185)

van Gigch, J. P. (1986). Modeling, Metamodeling, and Taxonomy of System Failures, I*EEE trans. on reliability*, vol. R-35, no. 2, 1986 June, pp(131-136)

van Gigch, J. P. (1991). *System design Modeling and Metamodeling,* Plenum, 0306437406, New York

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

Weick, K. E. & Sutcliffe, K. M. (2001). *Managing the Unexpected: Assuring High Performance in an Age of Complexity (J-B US non-Franchise Leadership),* Jossey-Bass, 0787956279, San Francisco



System of Systems

Edited by Dr. Adrian V. Gheorghe

ISBN 978-953-51-0101-7
Hard cover, 114 pages
Publisher InTech
Published online 02, March, 2012
Published in print edition March, 2012

The present book proposes and fosters discussion on the current applications in the field of system of systems, with emphasis on the implications of the fact that new developments and area of technical and non-technical applications are merging. The book aims to establish an effective platform for communication among various types of practitioners and theory developers involved in using the system thinking and systems engineering approaches at the scale of increased complexity and advancing computational solutions to such systems.

Takafumi Nakamura and Kyoichi Kijima (2012). System of System Failure: Meta Methodology to Prevent System Failures, System of Systems, Dr. Adrian V. Gheorghe (Ed.), ISBN: 978-953-51-0101-7, InTech, Available from: http://www.intechopen.com/books/system-of-systems/system-of-system-failure-meta-methodology-to-prevent-system-failures

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821