TOTAL SYSTEM INTERVENTION FOR SYSTEM FAILURES AND ITS APPLICATION TO ICT SYSTEMS

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

[1]Fujitsu FSAS Inc., Support Technology Group, Masonic 38 MT Building, 4-1-4, Shibakoen, Minato-ku, JAPAN, nakamura.takafu@jp.fujitsu.com

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

*Corresponding author

ABSTRACT

Total system intervention for system failure (TSI for FS) is proposed for preventing further occurrences of system failures. Total system intervention (TSI) is a meta-methodology in critical system thinking for managing complex and differing viewpoints. First, the authors introduce 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 “total system intervention for system failures (TSI for FS)”. TSI for FS identify the stakeholders in the failure using a matrix that shows for each stakeholder the entity and/or the factor that is thought to have caused the failure. This helps to clarify the stakeholders’ views and to identify stakeholders with opposing views. The SOSF meta-methodology 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 countermeasures and/or preventative measures to be applied. An application example in information and communication technologies engineering demonstrates that using the proposed meta-methodology as a critical system practice helps prevent future system failures by learning from previous system failures. Three actions were identified for preventing further system failures: close the gap between the stakeholders, introduce absolute goals and enlarge system boundary.

Key words: total system intervention, critical system thinking, critical system practice, system failure model, structuring methodology, double-loop learning, risk management

1. INTRODUCTION

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.

Weick and Sutcliffe (2001, p. 81) explained why traditional total quality management (TQM) has failed. “We interpret efforts by organizations to embrace the quality movement as the beginning of a broader interest in reliability and mindfulness. But some research shows that quality programs have led to only modest gains...this might be the result of incomplete adoption. But we would go even further, and argue that the reason for incomplete adoption is the necessary infrastructure for reliable practice…is not in place even where TQM success stories are the rule. The conclusion is consistent with W.E. Deming’s insistence that quality comes from broad-based organizational vigilance for problems other than those found through standard statistical control methods.”

There are six stages in reaching a fatal system failure. The second stage, or incubation period, is hard to identify due to the various side effects of quick fixes (Turner, 1997). Many side effects due to quick fixes of information and communication technologies (ICT) systems have been identified (Nakamura and Kijima, 2009a). There are two factors in particular that make it difficult to prevent ICT system failures: the lack of a common language for understanding system failures and the lack of a methodology for preventing future system failures. These shortcomings result in local optimization and the introduction of quick fixes as countermeasures. Habermas (1970, 1975, and 1984) argued that there are two fundamental conditions underpinning the sociological life of human beings: ‘work: technical interest’ and ‘interaction: practical interest’. Disagreements between individuals and groups are just as much a threat to the socio-cultural form of life as a failure to predict and control. The core idea of intervention methodologies is to accommodate multiple stakeholders and to identify the best methodology for restoring a failed system.

We propose using the “system of system failures (SOSF)” meta-methodology to provide a common language for understanding system failures among the various stakeholders. We also propose using “total system intervention for system failure (TSI for SF)” as a methodology for preventing future system failures of the same type. The SOSF meta-methodology and a stakeholder matrix are used within the TSI for SF methodology. Application examples of ICT systems were used to demonstrate that the TSI for SF methodology is effective.

2. SOSF META-METHODOLOGY AS A COMMON LANGUAGE

2.1 System of system failure

The proposed SOSF meta-methodology for covering all system failures models (Nakamura, Kijima, 2007, 2008b, 2009a) is derived from system of system methodologies (SOSM) (Jackson, 2003) and system failure classes. SOSM classifies the world of objects into two dimensions: systems and participants. The system dimension has two domains: simple and complex. The participant dimension has three domains: unitary, plural, and coercive. Therefore, SOSM classifies the world of objects into six (2 × 3) domains, and there is an appropriate methodology for each domain. SOSF complementarily covers the domains on the basis of the worldview to enable viewing of the objects system failures. SOSF uses four domains (excluding the coercive domain) from SOSM. On top of these four domains, we add a third dimension to identify the person or factor responsible for the system failure. To identify the root causes of failures, we classify system failures on the basis of the system boundary and the responsible system level introduced VSM model (Beer, 1979, 1981). Failures are classified in accordance with the following criteria (Nakamura, Kijima, 2008b, 2009ab).

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

Class 2 (Failure of interface): The root cause is outside the system boundary but is predictable in the design phase.

Class 3 (Failure of foresight): The root cause is outside the system boundary and is unpredictable in the design phase.

Figure 2.1 shows the SOSF meta-methodology space.

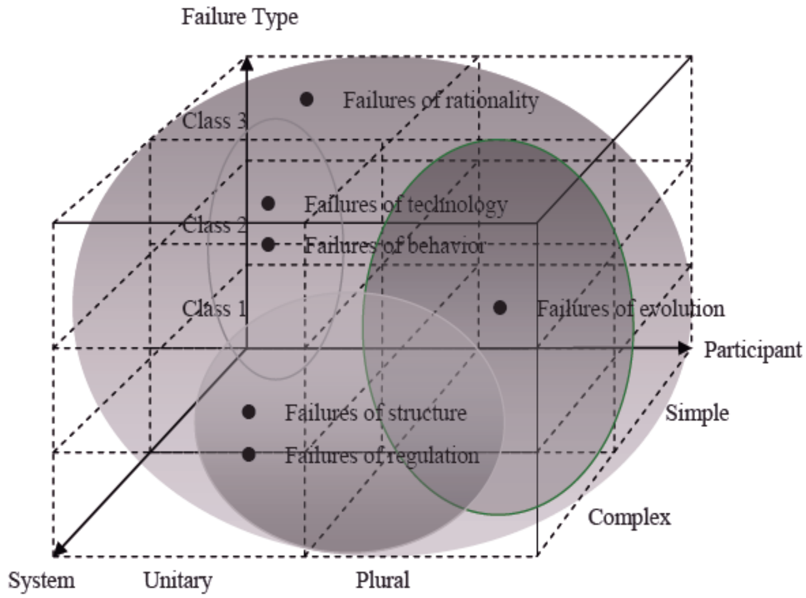


Fig. 2.1 SOSF meta-methodology space

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

2.2 Failure factor structuring methodology

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) promotes double-loop learning through viewing the system in a holistic way. Figure 2.2 shows a general overview of this methodology, and Table 2.1 lists the objectives for each phase of FFSM.

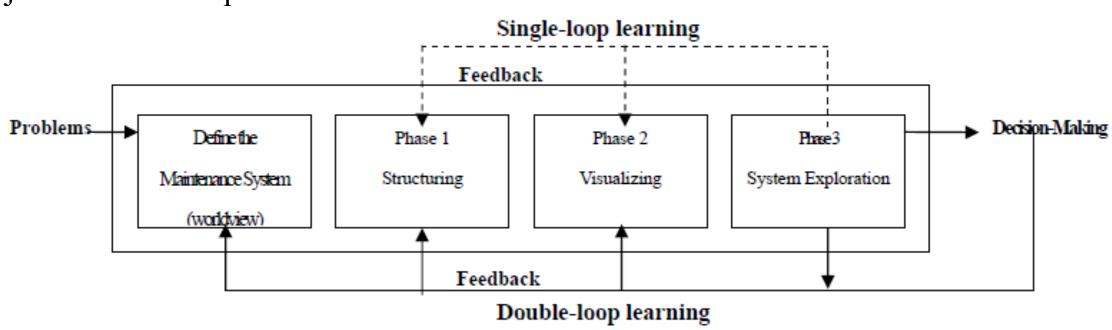


Fig. 2.2 General overview of FFSM

Table 2.1 Objectives for phases 1, 2, and 3 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

2.3 System failure dynamic model

We propose new nonlinear systematic 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. 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, p. 88). 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, 2004) and that side effects due to quick fixes are prevented.

2.4 Current methodologies

Two failure analysis methodologies are widely used: failure mode effect analysis (FMEA: IEC 60812) and fault-tree analysis (FTA: IEC 61025). FMEA deals with single-point failures by taking a bottom-up approach; it is presented as a rule in the form of tables. In contrast, FTA analyzes combinations of failures in a top-down manner, and the results are visually presented as a logic diagram. Both methodologies are mainly used in the design phase. However, they depend heavily on personal experience and knowledge. FTA in particular has a tendency to miss some failure modes in failure mode combinations, especially emergent failures.

2.5 Summary

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 Fig. 2.3.

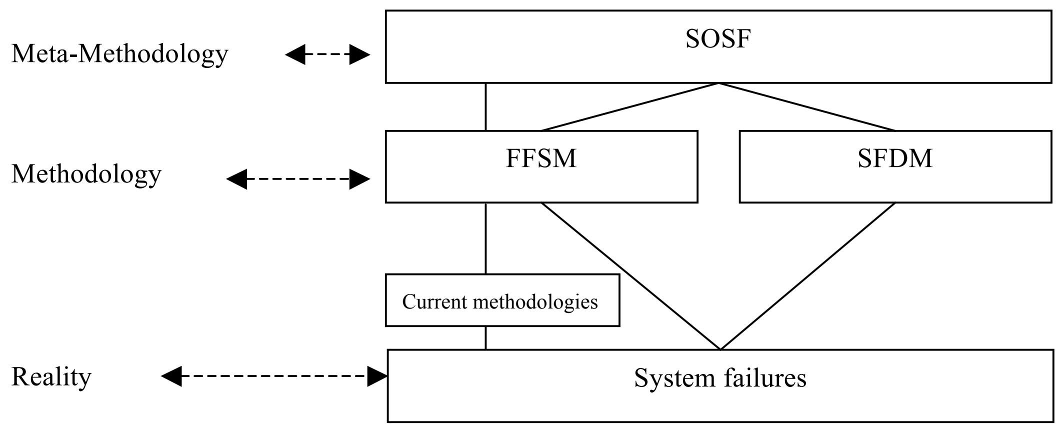


Fig. 2.3 Relationships among SOSF, FFSM, and SFDM

Table 2.2 summarizes the system failure models and related methodologies as well as the meta-methodology. Table 2.3 shows the methodology mapping onto SOSF space.

Table 2.2 Three system-failure models and their approach to management

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, Kijima, 2008a, 2009b)	
Systemic Model; Unrocking Boat Metaphor (Reason, 1997) Rasmussen's Gradients Model (1997)	Plural	Balance Variability	SFDM (Nakamura, Kijima, 2008b), Six Stages (Turner, 1997)	

Table 2.3 Methodology mapping to SOSF space

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

3. TSI FOR SF METHODOLOGY AS AN APPLICATION PROCEDURE

Flood and Jackson (1991) identified seven principles underpinning the TSI.

- Problem situations are too complicated to understand from one perspective, and the issues they throw up are too complex to tackle with quick fixes.
- Problem situations, and the concerns, issues, and problems they embody, should therefore be investigated from a variety of perspectives.

- Once the major issues and problems have been highlighted, a suitable systems methodology or methodologies must be identified to guide intervention.
- 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.
- 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.
- The TSI sets out a systematic cycle of inquiry with interaction back and forth between its three phases.
- Facilitators and participants are engaged at all stages of the TSI process.

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 and 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.

Phase1. Become aware of system failure
Phase2. Identify stakeholders
Phase3. Creativity: identify Metaphors

In the creativity phase, the many different possible views of organizations and their problems are recognized, and managers and analysts are 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.

Phase4. Choice: Select methodology using SOSF meta-methodology

In this phase, the metaphors generated in the creativity phase are mapped to the SOSF space (Nakamura and 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 2.2. Note that the SOSF meta-methodology is used not to deterministically prescribe which methodology to choose but to illuminate and inform that choice.

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. 3.1, 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.

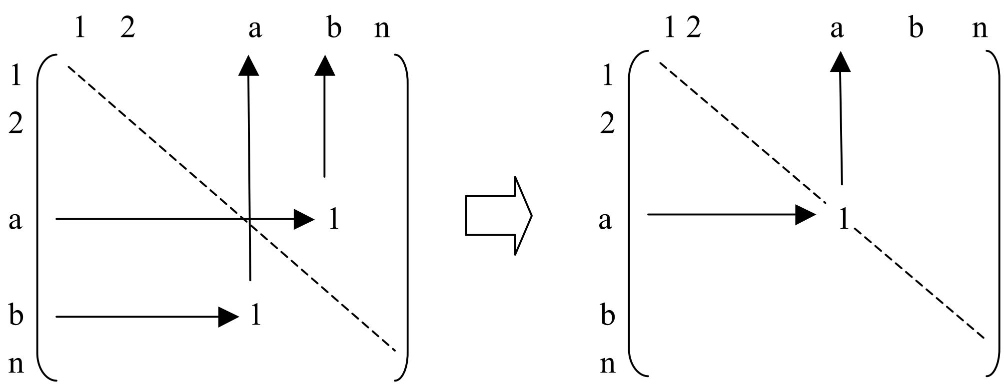


Fig. 3.1 Stakeholder matrix

Phase5. Implementation: Take action

In the implementation phase, methodologies are applied to produce change. In this phase, the selected methodology (FFSM or SFDM) is used in accordance with the complementary principles of TSI.

Phase6. Reflection: Acquire new learning

In the reflection phase, the intervention is evaluated and learning about the problem situation, the meta-methodology itself, the generic system methodologies, and the specific methods used is produced. The outcome is research findings that are used, for example, as feedback for improving earlier stages of the meta-methodology.

The metaphors, the SOSF meta-methodology, and related methodologies as a core concept of TSI for FS methodology are shown in Fig. 3.2.

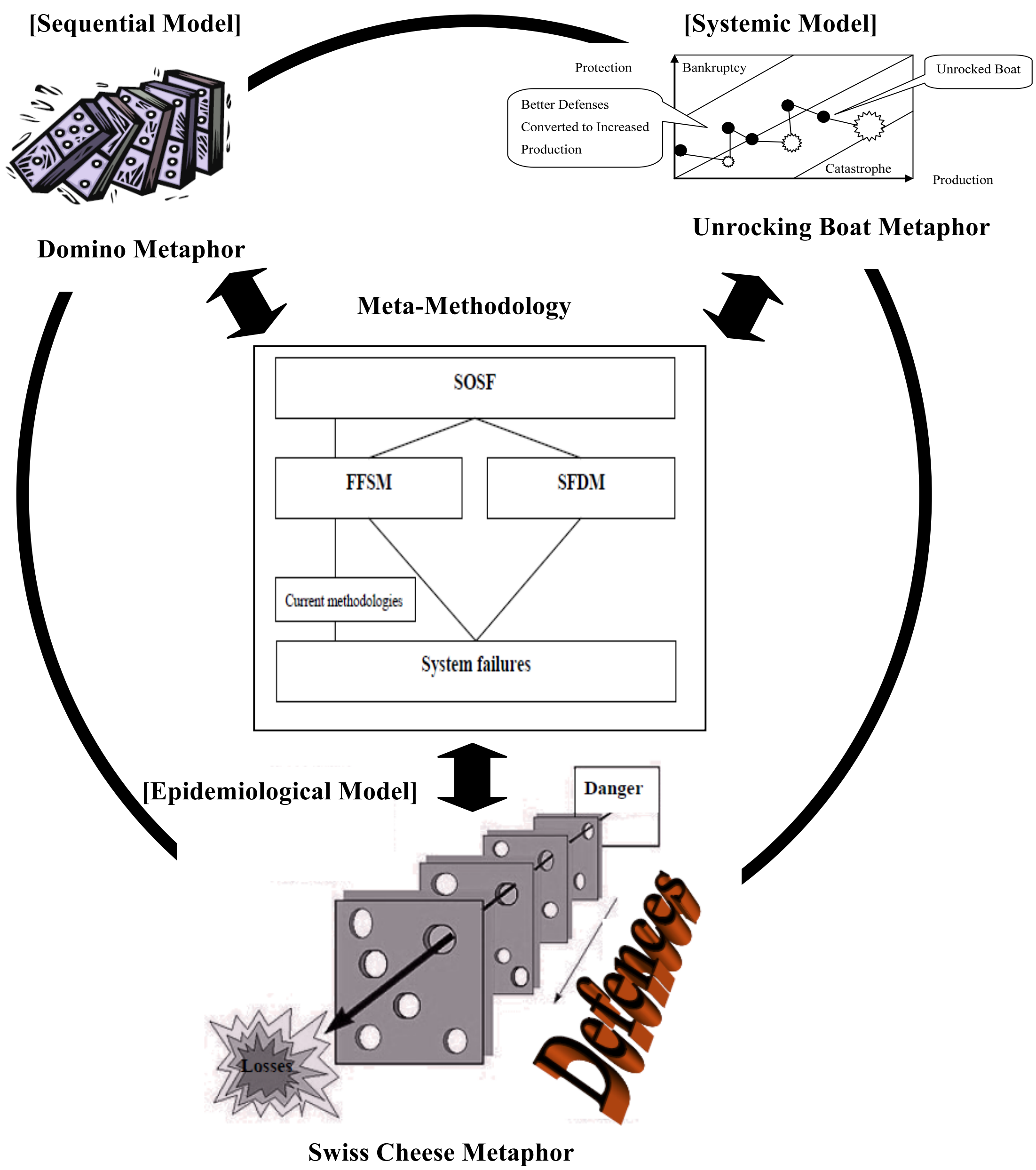


Fig. 3.2 TSI for FS methodology with metaphors, SOSF meta-methodology, and related methodologies

4. 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. 4.1. 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 disk 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. 4.2.

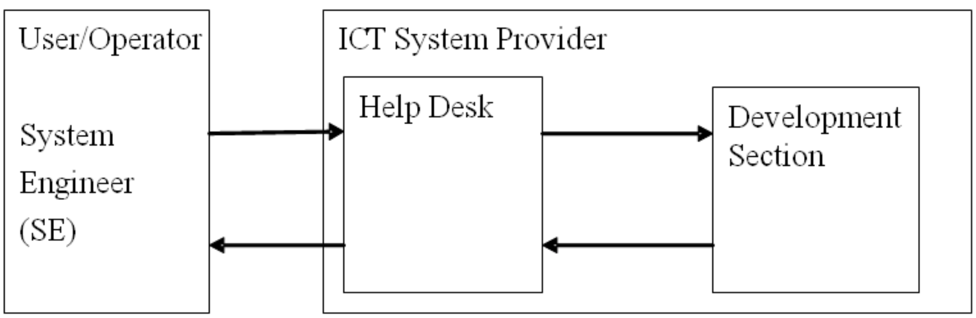
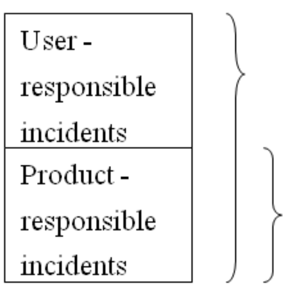


Fig. 4.1 Incident escalation procedure



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

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

Fig. 4.2 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.

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.

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.

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. 4.1). 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 (see Table 2.2) as appropriate for this situation (i.e., the organization is drifting through the environment between excessive economic gain and safety).

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

The stakeholder opinions are clarified using the stakeholder matrix in order to identify stakeholders with opposing views. As shown in Table 4.1, the SE and development section have opposing views. The Help desk claims that the SE made an error in operation.

Table 4.1 Stakeholder matrix

	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.	—	—

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).

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. 4.3) 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. 4.3. 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, which increases the number of Class 1 failures. 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.

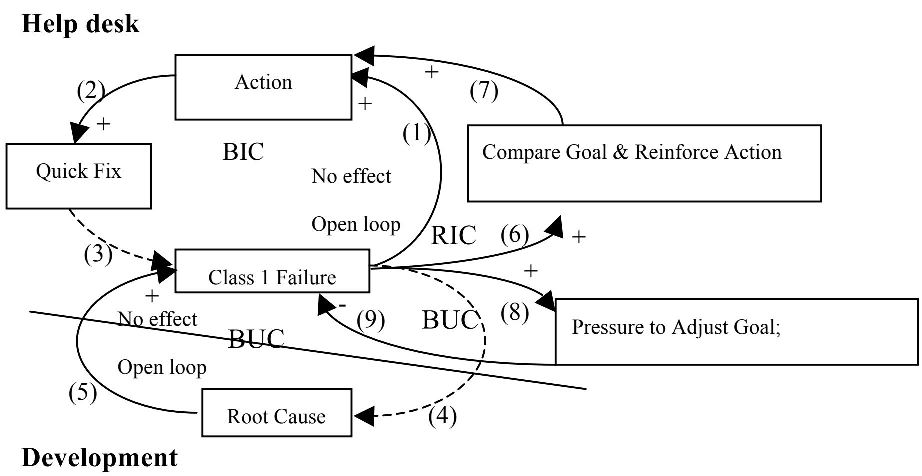


Fig. 4.3 Misunderstanding system failure archetype

The archetype shown in Fig. 4.3 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. 4.3. 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.

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. 4.3). 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. 4.3. 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.

Fix that fails archetype (side effect)

The source of the failure is outside the help desk's system boundary. Figure 4.4 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 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. 4.4. 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.

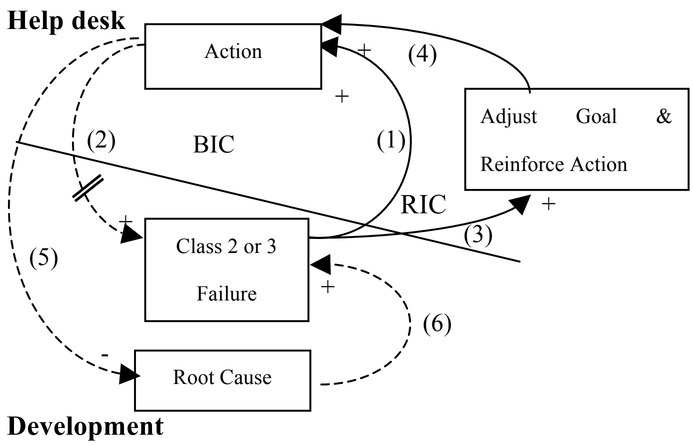


Fig. 4.4 Fix that fails archetype (side effect)

In this application example, the stakeholders can understand the situation holistically using the SOSF meta-methodology. The user thinks these errors are not operation-related but product-related. Conversely, the development section thinks they are operation-related. Therefore, the user insists that they are Class 3 failures of evolution in complex and plural domains in SOSF. Conversely, the development section insists that they are Class 1 failures of behavior in a simple and unitary domain. Figure 4.5 illustrates this situation, and Fig. 4.6 illustrates the consolidated SOSF space showing all stakeholder opinions.

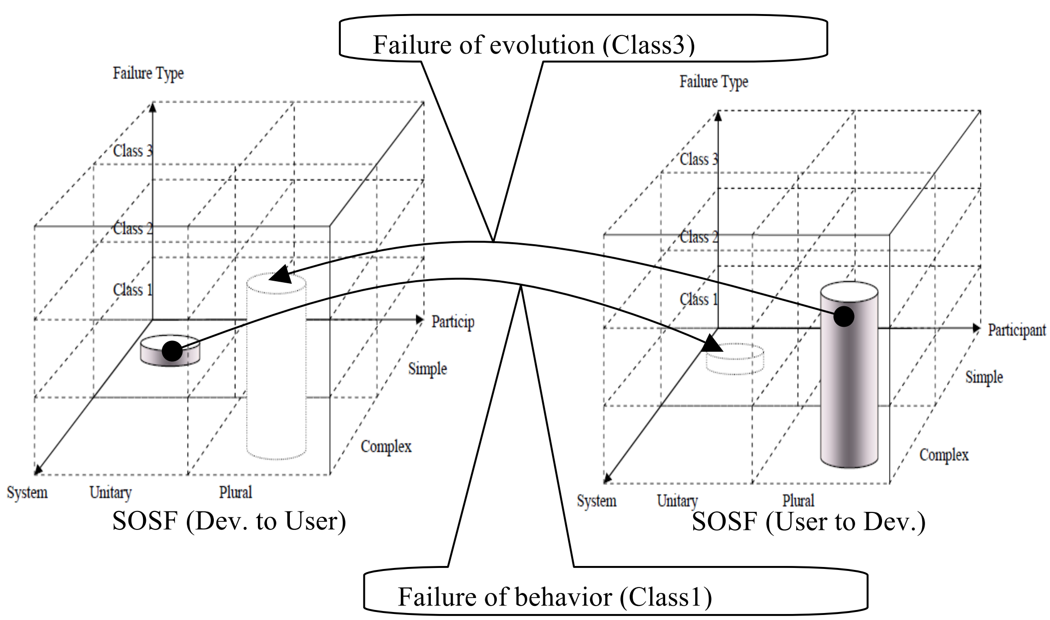


Fig. 4.5 Simple-Unitary (Class 1) vs. Complex-Plural (Class 3)

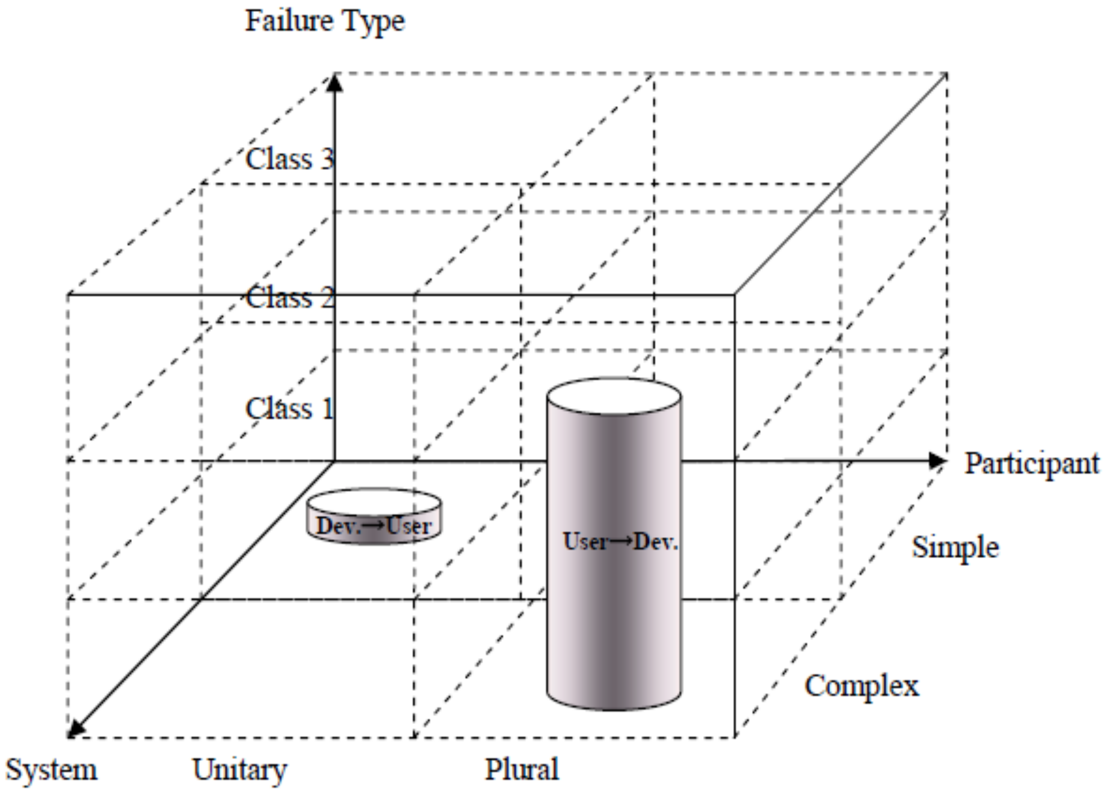


Fig. 4.6 Consolidated SOSF space

Phase 5. Implementation: Take action

After the debate and discussion, the stakeholders reached the conclusion shown in Table 4.2.

Table 4.2 Clarify stakeholder opinions using matrix

	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.

The SQA section analyzed the user-related incidents and, as illustrated in Fig. 4.7, 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. 4.7. 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. 4.8.

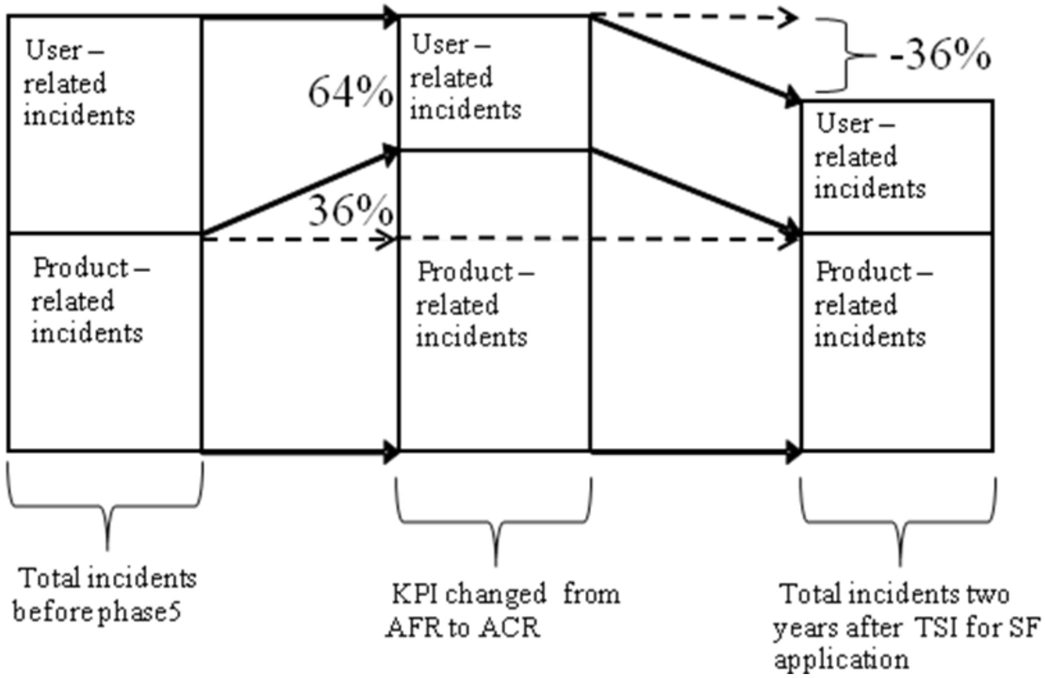


Fig. 4.7 Incidents transition over two-year period

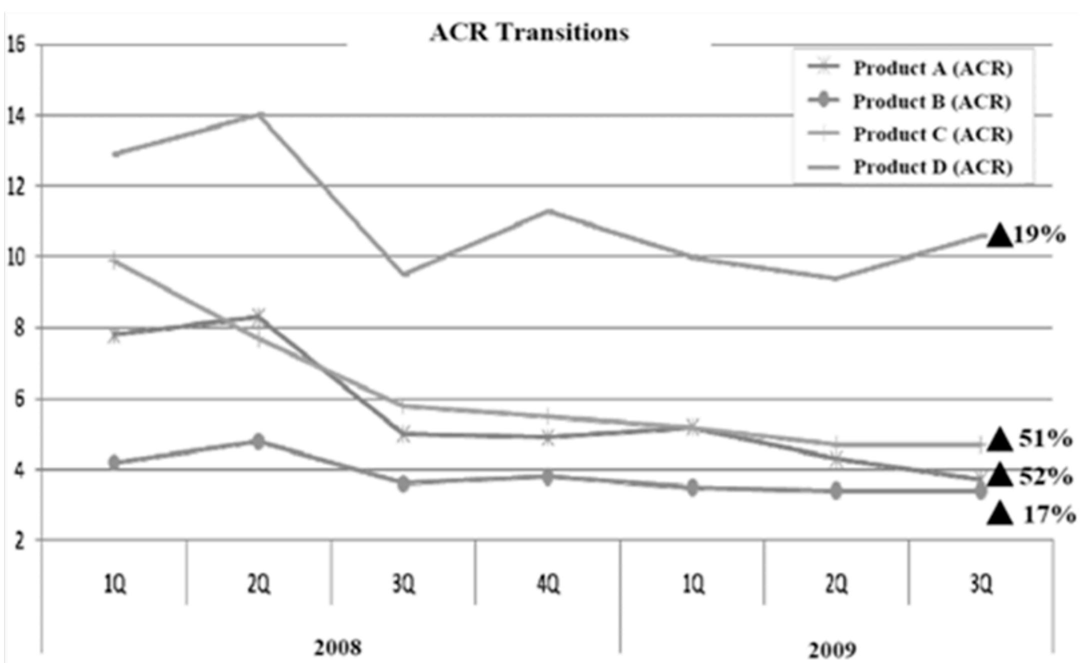


Fig. 4.8 ACR transitions

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.

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 made throughout the discussion and debate to close the responsibility gap. Application of this solution to the scenario shown in Fig. 4.3, misunderstanding system failure archetype, is illustrated in Fig. 4.9. 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.

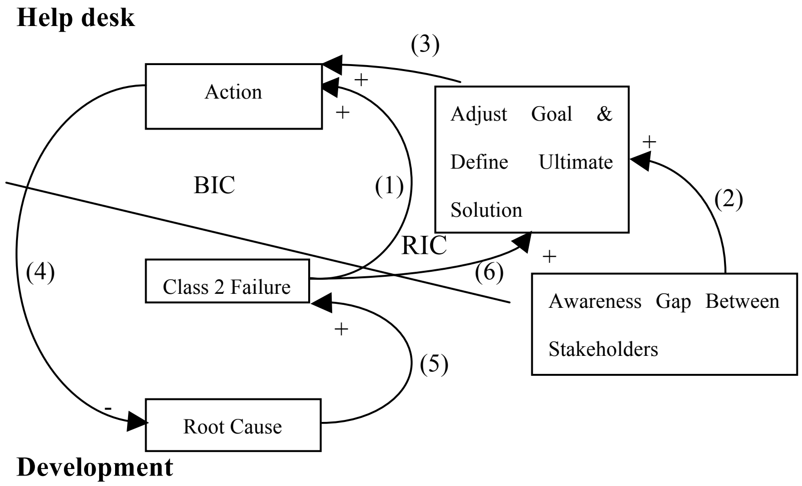


Fig. 4.9 Double-loop learning for Class 2 failure (solution)

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. 4.10. 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.

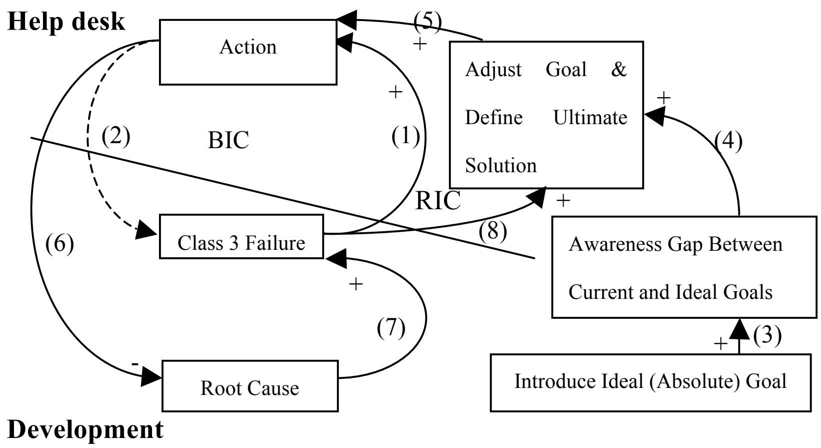


Fig. 4.10 Double-loop learning for Class 3 failure (solution).

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

The solution for this archetype is to raise the viewpoint of the problem (Fig. 4.4). 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. 4.11. 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).

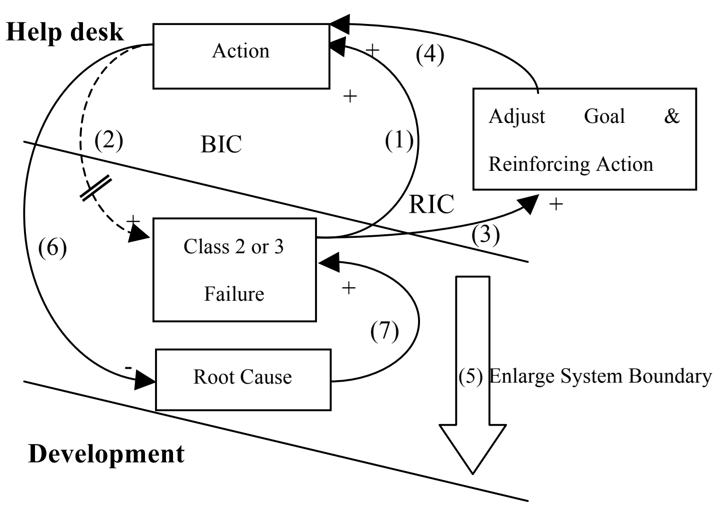


Fig. 4.11 Fix that fails archetype (side effect)

Figure 4.12 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 Fix that fails archetypes).

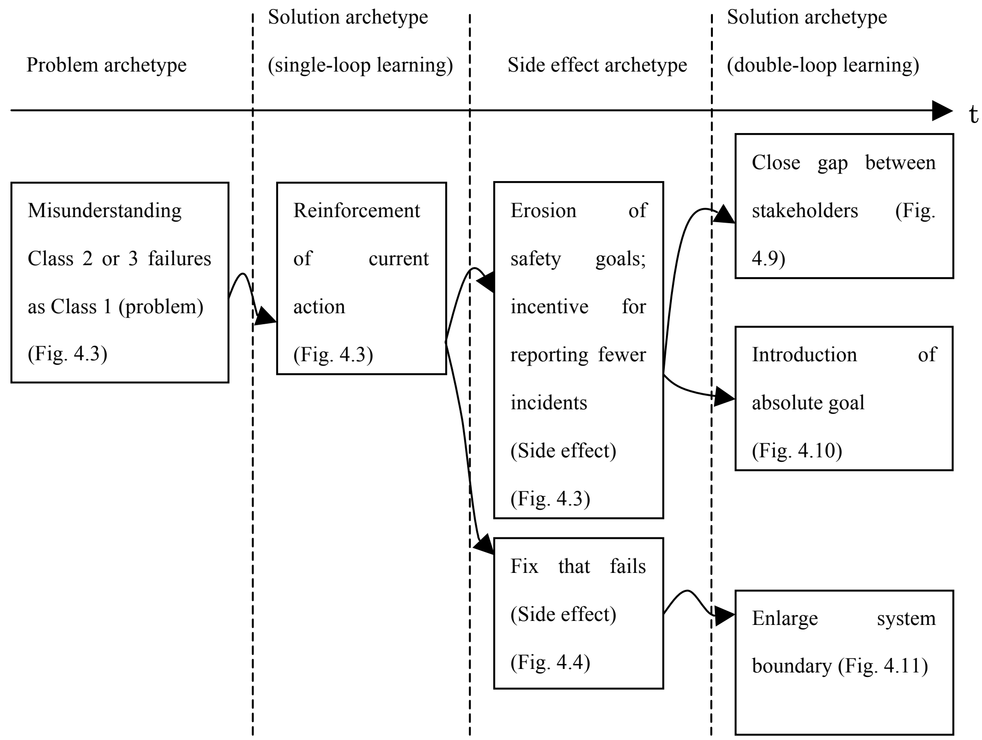


Fig. 4.12 Problem and solution archetypes in engineering system failures through time.

5. 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 4. 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' (Habermas, 1907, 1975, 1984). 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.

⇨ 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.

⇨ 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.

⇨ 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.

REFERENCES

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

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

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

Habermas, J. (1970). Knowledge and interest. In: D. Emmet and A. MacIntyre (eds.), Sociological Theory and Philosophical Analysis (pp. 36–54). Macmillan: London.

Habermas, J. (1975). Legitimation Crisis. Beacon Press: Boston.

Habermas, J. (1984). Reason and the Rationalization of Society. Beacon Press: Boston.

Heinrich, H.W., Petersen, D., and Roos, N. 1989. Industrial Accident Prevention: A

Safety Management Approach. 5th ed. McGraw-Hill: New York.
IEC 60812 (2006). Procedure for failure mode and effect analysis (FMEA)
IEC 61025 (2006). Fault tree analysis (FTA)
Jackson, M.C. (2003). Systems Thinking: Creative Holism for Managers. John Wiley & Sons: London and New York.
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: 647-657.
Nakamura, T., Kijima, K. (2007). Meta system methodology to prevent system failures. Proceedings of the 51st Annual Meeting of the ISSS in 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: 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 in 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: 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 in Brisbane (Jul. 2009).
Perrow, C. (1999). Normal Accidents: Living with High-Risk Technologies. Princeton Paperbacks: New York.
Rasmussen, J. (1997). Risk Management in a dynamic society: a modeling problem. *Safety Science* vol. 27, no. 2/3: 183–213.
Reason, J. (1997). Managing the Risk of Organizational Accident. Ashgate Pub Ltd.
Reason, J., Hobbs, A. (2004). Managing Maintenance Error: A Practical Guide. Ashgate Pub Ltd.
Turner, B.A., Pidgeon, N.F. (1997). Man-Made Disasters 2nd edition. Butterworth-Heinemann: UK.
Van Gigch, J. P. (1991). System Design Modeling and Metamodeling. Plenum: New York.
Weick K.E., Sutcliffe K.M., (2001). Managing the Unexpected: Assuring High Performance in an Age of Complexity (J-B US non-Franchise Leadership).