Defect Prevention Matrix Analysis by Generative AI

Shuichiro Yamamoto
Department of Information Technology
IPUT in Nagoya
4-27-1 Meieki Nakamura-ku, Nagoya, Japan
yamamoto.shuichiro@gmail.com

***Abstract*—To clarify the integrity of business processes, we propose a business process diagram that describes six aspects: input, output, acceptance conditions, resource conditions, exception conditions, and decision conditions. By separating exception conditions from output, defect prevention diagrams have the advantage of being able to detect defects and extract corresponding exception handling knowledge. However, methods for reviewing defect prevention diagrams have not been clear. In this paper, we define a process relationship matrix and present a review procedure to prevent business process defects. We also demonstrate through application experiments that the review procedure can be automated using generative AI and that six aspects of issues can be detected for each process. The main results of this paper are a business process review method using a process relationship matrix and a prompt template that automates this process using generative AI.**

Keywords- Business Process Completeness; Exception; Knowledge Management; Defect Prevention; Process Relationship Matrix; Generative AI.

I. INTRODUCTION

First, we note that this paper is an extension of the work described in [1]. In this paper, we demonstrate that the procedure proposed in the international conference paper can be automated by defining it as a prompt for generative AI, and we also add application examples to demonstrate its effectiveness. Furthermore, we clarify that the purpose of the proposed method is to detect defects in production and operational processes, not to analyze defects in product shape.

Business process modeling is necessary for corporate digital transformation. While notational methods for defining business models have been proposed to date, they pose the following problems.

If the purpose and conditions for achieving a business process cannot be clearly described, the goals that stakeholders must agree on are unclear, making it impossible to confirm the completeness of the business process.

The execution order of business processes alone not only makes the resources required for process execution and completion conditions unclear. Also, it is impossible to control priority when inputs from multiple preceding processes conflict. Furthermore, because exceptions that occur in the business process are not clearly defined, it is impossible to design appropriate responses.

For this reason, defect prevention diagrams have been proposed. Defect prevention diagrams address the above issues by describing six aspects of each business process: input, output, acceptance conditions, resource conditions, decision conditions, and exception conditions. Furthermore, the validity of a defect prevention diagram can be confirmed using a business process relationship matrix, which clearly defines the relationships between business processes.

This paper presents an analytical method using a defect prevention matrix to review business processes and a prompt template that automates the analysis using generative AI.

The rest of this paper is organized as follows. Section II describes related work. Section III describes the defect prevention matrix analysis. Section IV explains experimental case study. Section V presents discussions. The conclusion closes the article.

II. RELATED WORK

Related studies on Ji-Koutei-Kanketsu (JKK), Knowledge transfer, Business Process Modeling (BPM), Defect Prevention Analysis (DPA) and Functional Resonance Analysis Method (FRAM) are explained below.

A. Ji Koutei Kanketsu

In the production process, there is a misconception that local optimization is necessary, as long as one's own process is fine, and that unnecessary problems shall not be introduced to one's department. If a problem is discovered at the final stage of development, the design cannot be modified, or the basic structure of the product cannot be changed. Therefore, comprehensive product design and manufacturing are required throughout the entire production process. Ji-Koutei-Kanketsu (JKK) is a method that optimizes the entire production process, not just a specific process. The Japanese words Ji, Koutei, and Kanketsu [2] are self, process, and completion, respectively.

To introduce JKK, it is necessary to define not only business procedures that define the flow of work, but also requirements organization sheets that define business requirements. The requirements organization sheet consists of fields of the necessary items/information, business inputs, and business outputs for each business process. The necessary item and information field clarifies the input, tools, methods, capabilities/authority, and reasons as conditions for the quality of the product. The input field describes the receiving criteria, such as when, where, and what. The output field describes where to sink, by when, and what to produce. The criteria field describes criteria for determining that "the output of the process is good."

JKK's production processes can also be seen as business processes. JKK clarifies the completeness conditions for each business process element. The requirement organization sheet is an essential feature of JKK.

B. Knowledge transfer

In order to transfer a company's experiential knowledge, it is necessary to clarify business processes. For this reason, methods for clarifying business processes have been proposed for knowledge transfer.

From a knowledge perspective, processes need to be defined to provide appropriate knowledge for tasks in an organization's operational business processes. In addition, knowledge must be extracted for the long-term growth, development, and competitiveness of companies. However, unless valuable knowledge within an organization is externalized or formalized, it cannot be used by other employees and disappears from the company. Therefore, Knowledge management shall be established using Business Process Modeling (BPM). Salvadorinhoa and Teixeira [3] pointed that BPM can not only help organizations improve their Industry 4.0 environment but also facilitate knowledge acquisition and distribution.

C. Business Process Modeling

Ore et al. [4] proposed a Self-managed organization based on Business Process Management. They showed a need for the business process management approach, which would manage the need for keeping critical business processes continuity and self-managed way of working of autonomous teams.

As long as the digitalization of business is promoted, business process documentation becomes vital for business process continuity. The digitalization re-constructs the traditional business processes into new digitalized business processes [5]. For example, the Digital Balanced Scorecard (DBSC) [6] consists of digital business processes.

There are many Business Model notations, including Business Process Models. Yamamoto [7] compared the representation capability of Business Model notations by defining fifteen key features of these notations with five interrogatives.

Leonard and Swap [8] defined deep smart as the expertise that allows experts to instantly grasp complex situations and make quick and wise decisions to deal with real problems. That is, deep smart is strong expertise formed by beliefs and social influences that can generate insights based on tacit knowledge grounded in direct experience. For example, in production process design, the problem is how to transfer defect investigation knowledge from experienced workers to beginners. An example of deep smart is the failure investigation knowledge that experienced engineers have. Leonard and Swap pointed out the importance of acquiring empirical knowledge through experimental learning. However, no concrete experimental learning method has been clarified. In addition, they have not clarified the knowledge representation of deep smarts. If deep smart cannot be expressed, it remains tacit knowledge, and deep smart knowledge transfer from experts to beginners is individual and difficult to spread horizontally.

As a technique for improving production processes in the manufacturing industry, Mono-Koto-Bun-seki (MKB) (in Japanese) has been proposed [9]. Mono, Koto, and Bun-seki mean Entity, Process, and Analysis, respectively. By treating objects such as materials and products as “entities” and the series of activities that make products from materials as "process," MKB can analyze the production process, discover waste, and optimize it.

Yamamoto and Fujimoto [10] proposed the Production Knowledge Chart (PKC) that expresses the production process to acquire the empirical knowledge necessary for investigating defects in manufacturing processes.

Object Process Methodology (OPM) proposed by Dori includes Object and Process [11][12]. For example, the aircraft design OPM has a Stakeholder Needs Set, Assumptions and Constraints Sets, and Requirements as Objects. There are three types of Processes: Defining, Realizing, and Implementing. In addition, physical Objects include Aircraft, System, Item, and Item component.

D. Functional Resonance Analysis Method

Functional Resonance Analysis Method (FRAM) [13] has been used to analyze complex functional resonances of socio-technical systems through functional networks. The FRAM function is defined by hexagonal nodes with six sides. These sides correspond to six aspects which are Input, Output, Time, Control, Resource, and Precondition. The output side of a function can be connected to the other five sides of other functions. FRAM provides useful means for safety analysis. Possible aspect relationships are <O, I>, <O, T>, <O, C>, <O, R>, and <O, P>. Here, <X, Y> is where X and Y are functional aspects.

The following three types of FRAM matrix representations have been proposed.

Lundberg and Woltjer [14] proposed a Resilience Analysis Matrix (RAM) to visualize functional dependencies between complex systems. RAM is a square matrix that shows the propagation relationship between functions. The size of RAM is the number of functions in FRAM. Element (i, j) of RAM indicates that some aspect of function i is propagated from the output of function j. The diagonal element (i, i) of RAM is the output of function x.

Patriarca et al. [15] proposed another square matrix composed of aspect combinations of FRAM functions. If there are n couplings in FRAM, RAM is defined as an $n \times n$ square matrix. The value of RAM (i,j) is 1 or 0.

Functional Aspect Resonance Matrix (FARM) is a non-square matrix that shows the propagation relationship between the output of a function and other aspects. The number of rows in FARM is the number of output sides of the function that are propagated to other functions in FRAM. The column size of FARM is the number of sides of a function that are connected to the output sides of other functions. Element (i, j) of FARM indicates that some functional surface j is propagated from the output of function i. In general, the number of rows and columns in FARM is not equal, so there are no diagonal elements. The equivalence of the above three matrices has been shown by Yamamoto [16].

E. *Defect detection by visual feature analysis*

A blog post [17] from dataspa.ai notes that visual inspection is a critical process in the manufacturing industry, where minute surface defects (scratches, cracks, peeling, etc.) have a significant impact on product quality. However, manual inspection is subjective and unscalable. Therefore, utilizing generative AI and synthetic data to artificially generate defect images can address data shortages. The synthetic data improves model accuracy and reliability, while reducing implementation costs.

A research paper [18] by Singh et al. comprehensively reviews synthetic image generation technologies to improve the performance of deep learning models for surface defect detection in the manufacturing industry. Deep learning models require large amounts of training data, but collecting defect images on-site is difficult and costly. On the other hand, CG-based approaches using CAD models and renderings lack the feasibility of defect generation.

These papers highlight the common challenge of data shortages in detection in the manufacturing industry, and the promise of synthetic data generation using generative AI as a solution.

However, the defect detection in manufacturing that these two documents focus on deals with "physical and visual features" such as images and sensor data, and defines defects as abnormalities in shape, color, or pattern.

However, defects in production and business processes arise from abstract and logical structures such as business flows, decision-making, and communication. Because these cannot be expressed as images or simple numerical patterns, they cannot be addressed by methods that rely on image generation or visual feature learning.

III. DEFECT PREVENTION MATRIX ANALYSIS

A. *Defect Prevention Diagram*

A defect prevention diagram consists of business processes and flow relationships between business processes. In a business process, input, output, accepting conditions, resource conditions, judgment conditions, and exception conditions are clarified. Flow relationships include flows from output to input and flows from exception conditions to input, resource conditions, and judgment conditions.

The Input describes the trigger and information for starting an action. The Output describes the response and information as a result of the action. Accepting conditions describe the conditions for executing an action. Resource conditions describe the people, equipment, information, and activities required to output the action results. Judgment conditions describe the criteria for outputting the action results. Exception conditions describe the conditions under which output cannot be generated because the receiving conditions, resource conditions, and judgment conditions are not met.

Figure 1 shows the defect prevention diagram process element.

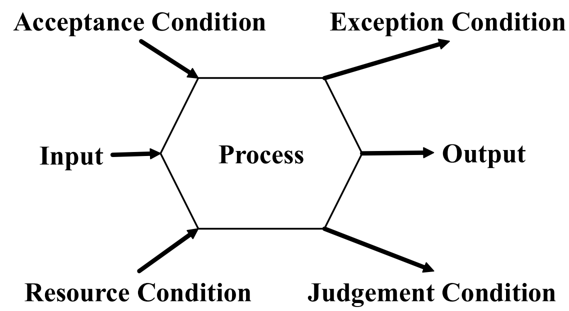

Figure 1. Defect prevention diagram process.

B. *Defect Prevention Matrix*

For the business process P that constitutes the defect prevention diagram D, the defect prevention matrix M can be defined as follows.

In Table I, S and T are either the receiving condition A, the resource condition R, or the judgment condition J. If S and T are omitted, they are taken to be the relationship to the input of the target process.

TABLE I. DEFECT PREVENTION MATRIX

	X	Y
X	Goal of X	X to S: Y Relationship
Y	Y to T: X Relationship	Goal of Y

The diagonal element M (X, X) describes the purpose of business process X. The off-diagonal element M (X, Y) describes the connection flow from business process X to either the input, receiving condition, resource condition, or judgment condition of Y.

The defect prevention matrix can be used to comprehensively check the connection flow between business processes that make up the defect prevention diagram. For example, the transitive closure of the defect prevention matrix can define a set of connection relationships for business processes. The set of connection relationships for X in Table I is $\Sigma_{k=1,n} (R_{xy} \cdot R_{yx})^k$. R_{xy} is the relationship from X to S: Y, and R_{yx} is the relationship from Y to T:X.

Similarly, the set of connection relationships for Y in Table I is $\Sigma_{k=1,n} (Ryx \cdot Rxy)^k$.

The defect prevention matrix is used to identify defects caused by the flow relationship among processes.

The scalability of the matrix approach depends on the complexity of the number of relations between processes. The approach can be adaptable for any business process relationships by using matrix representation.

C. Process Review List

The process review checklist (PRC) is defined as issues of concern for six aspects, as shown in the following template in D.

D. Prompt template Defect Prevention Matrix Analysis

The prompt template is designed as follows:

> The business process relationship matrix M can be defined as follows.
> The diagonal element M (X, X) describes the purpose of business process X. The off-diagonal element M (X, Y) represents the connection flow from business process X to either the input, receiving condition, resource condition, or judgment condition of Y.
> Develop M for the following < descriptions>.
> <Body of description>
> By using the following checklist, identify defects in the M processes.
> Checklist:
> The process review checklist is defined as issues of concern for the six aspects, as follows.
> [<Process name>]
> [Input] issues with input labels
> [Accepting condition] issues with accepting arrow labels
> [Resource condition] issues on resource arrow labels
> [Judgement condition] issues with judgement arrow labels
> [Output] issues with output arrow labels
> [Exception condition] issues on exception arrow labels

IV. EXPERIMENTAL STUDY

A. Shinkansen Bogie Inspection System

The Shinkansen bogie inspection system is a system implemented by JR Central to inspect Shinkansen bogies in order to maintain safe Shinkansen operation. The main components of the Shinkansen bogie inspection system are (1) the Shinkansen bogie to be inspected, (2) the maintenance workers who check for abnormalities in the bogie, (3) the controller who issues orders for the bogie inspection, and (4) the supervisor who manages the entire train operation management process.

The above description of the Shinkansen bogie inspection system is provided to the prompt template of the Defect Prevention Matrix Analysis. Then the following response is generated.

To develop the **Defect Prevention Matrix (M)** for the **Shinkansen bogie inspection system**, Human first identify the key business processes based on the components AI listed:

- **Bogie Inspection** (performed on the Shinkansen bogie)
- **Maintenance Work** (performed by maintenance workers)
- **Inspection Control** (managed by the controller)
- **Operation Supervision** (managed by the supervisor)

TABLE II. DEFECT PREVENTION MATRIX M

M (X, Y)	Bogie Inspection (BI)	Maintenance Work (MW)	Inspection Control (IC)	Operation Supervision (OS)
Bogie Inspection (BI)	Purpose: To assess the physical condition of the Shinkansen bogie for abnormalities.	Provides inspection results as input for maintenance decisions.	Receives inspection orders from IC.	Supplies inspection data for operational oversight.
Maintenance Work (MW)	Uses inspection data to perform necessary repairs or adjustments.	Purpose: To correct any abnormalities found in the bogie.	Receives task assignments based on IC decisions.	Reports maintenance status for operational tracking.
Inspection Control (IC)	Issues inspection orders and schedules.	Sends instructions and resource allocations to MW.	Purpose: To manage and coordinate inspection activities.	Receives strategic directives and constraints from OS.
Operation Supervision (OS)	Uses inspection and maintenance data to oversee train safety and scheduling.	Evaluates maintenance outcomes to ensure operational readiness.	Provides high-level directives and judgment conditions to IC.	Purpose: To ensure safe and efficient train operation through oversight of all processes.

Generative AI defines the matrix **M** as shown in Table II, where:

- **Diagonal elements M (X, X)** describe the **purpose** of each process.
- **Off-diagonal elements of M (X, Y)** describe the **connection flow** from process X to process Y (e.g., input, receiving condition, resource condition, or judgment condition).

To identify **defects** in the business processes of the Shinkansen bogie inspection system using the **process review checklist**, generative AI evaluates each process across the six aspects. Issues detected for business processes by generative AI are shown in Tables III through VI, respectively. Each Table corresponding to business processes shows issues for six aspects.

TABLE III. BOGIE INSPECTION (BI) TABLE

Aspect	Issues
Input	Lack of standardized data formats from previous inspections. Incomplete historical maintenance records.
Accepting condition	No clear criteria for when a bogie is ready for inspection (e.g., post-trip, time-based). Manual confirmation may lead to delays or errors.
Resource Condition	Limited availability of inspection tools or sensors. Dependency on human inspectors without automation support.
Judgement condition	Subjective assessment by inspectors may vary. No AI or rule-based system to assist in anomaly detection.

Output	Inspection results may be inconsistently documented. Lack of integration with maintenance scheduling systems.
Exception condition	No protocol for handling unexpected bogie conditions (e.g., severe damage). Delays in reporting critical findings to supervisors.

TABLE IV. MAINTENANCE WORK CHECK

Aspect	Issues
Input	Inspection data may be unclear or lack actionable recommendations. Missing parts or repair history.
Accepting condition	No automated system to verify readiness for maintenance. Manual task assignment may cause bottlenecks.
Resource Condition	Shortage of skilled workers or spare parts. Equipment downtime.
Judgement condition	Decisions on repair priority may lack standardization. No predictive maintenance model in place.
Output	Maintenance logs may not be digitized or standardized. No feedback loop to inspection or control processes.
Exception condition	Emergency repairs may not follow standard procedures. Lack of escalation protocol for unresolved issues.

TABLE V. INSPECTION CONTROL CHECK

Aspect	Issues
Input	Incomplete data from operation supervision or previous inspections. Lack of real-time bogie status.
Accepting condition	No validation of inspection readiness. Manual scheduling is prone to errors.
Resource Condition	Limited access to real-time data or analytics. Over-reliance on manual coordination.
Judgement condition	No decision support system for prioritizing inspections. Lack of integration with train operation schedules.
Output	Orders may be delayed or miscommunicated. No audit trail for issued commands.
Exception condition	No contingency plan for inspection delays. Inadequate handling of conflicting inspection schedules.

TABLE VI. OPERATION SUPERVISION (OS) CHECK

Aspect	Issues
Input	Delayed or incomplete reports from inspection and maintenance. No real-time dashboard for system status.
Accepting condition	No automated alerts for critical issues. Manual review of reports may cause oversight.
Resource Condition	Limited access to integrated system data. Overload of supervisory responsibilities.
Judgement condition	Decisions may be reactive rather than predictive. No simulation tools for operational impact analysis.
Output	Strategic directives may lack clarity or timeliness. No feedback mechanism to lower-level processes.
Exception condition	No protocol for emergency train operation decisions. Lack of coordination during system-wide disruptions.

B. Insulin Pump Control System

Next, we applied and reviewed the method to an insulin pump control system [19].

Insulin pump requirements sentences are as follows;

R1) Insulin pump control software manages sensors implanted in the patient to measure blood parameters proportional to blood glucose levels.

R2) Blood parameters are sent to the pump controller.

R3) The controller calculates the sugar content and amount of insulin needed.

R4) The controller sends a signal to a small pump to administer insulin via a needle implanted in the patient.

R5) Insulin pump delivers 1 unit of insulin in response to 1 unit pulse from controller.

By using the Defect Prevention Matrix Analysis template, Generative AI breaks down the requirements into identifiable business processes:

BP1) Measure Blood Parameters (from R1)

BP2) Send Blood Parameters (from R2)

BP3) Calculate Insulin Needs (from R3)

BP4) Send Insulin Administration Signal (from R4)

BP5) Administer Insulin (from R5)

The key changes extracted are as follows.

a) Explicit Inputs and Resources: More precise descriptions of what each process receives and what it uses to operate.

b) Clearer Judgment Conditions: Adding checks or criteria that must be met for a process to proceed or complete successfully.

c) Crucial Exception Conditions: Highlighting potential failure points that require handling, essential for medical devices.

Table VII shows the elements of the Defect Prevention Matrix for the Insulin pump requirements. The matrix is shown in the form of purpose, conditions and flow columns. The matrix is shown in the form of purpose, conditions and flow columns. The purposes and conditions of diagonal elements are described in the raw. The off-diagonal elements are represented in flow columns. The form is selected by the sake of space efficiency.

TABLE VII. DEFECT PREVENTION MATRIX OF INSULIN PUMP REQUIREMENTS

	PURPOSE	CONDITIONS	FLOW
BP1	Continuously measures blood parameters using implanted sensors.	**Resources:** Functional implanted sensor. **Exceptions:** Sensor malfunction, disconnection, erroneous readings, inability to obtain a reading.	**Flow to BP2:** Measured raw blood parameter data sent to the pump controller.
BP2	Transmits the measured raw blood parameter data from the sensor to the pump controller	**Resources:** Operational communication module/channel (sensor to controller) **Exceptions:** Communication failure (signal loss, interference), data corruption during transmission.	**Flow to BP3:** Valid and complete blood parameter data successfully received from the sensor.
BP3	Processes received blood parameter data to calculate current glucose levels and determine the precise amount of insulin required.	**Resources:** Pump controller's processing unit, pre-programmed calculation algorithm. **Judgment Condition:** Calculated glucose level is within processing range; Calculated insulin amount is within pre-defined safe dosage limits	**Flow to BP4:** Insulin administration signal containing the calculated dosage amount, ready for transmission to the pump.

		Exceptions: Calculation error (e.g., invalid input, algorithm failure), calculated insulin amount is dangerously high or low, inability to determine a safe dose	
BP4	Transmits the calculated insulin administration signal from the controller to the insulin pump mechanism.	**Resources:** Operational communication module/channel (controller to pump). **Exceptions:** Signal transmission failure to the pump, signal corruption, non-receipt by the pump	**Flow to BP5:** Valid insulin administration signal/pulse successfully received from the controller.
BP5	Mechanically delivers the precise amount of insulin into the patient's body via the implanted needle based on the received signal.	**Resources:** Functional small pump mechanism, implanted needle/infusion set, **sufficient insulin in reservoir.** **Judgment Condition:** Sufficient insulin available in reservoir; Pump mechanism is operational (no blockages, sufficient battery); Needle/infusion set is properly functioning (not blocked/dislodged). **Exceptions:** Low or empty insulin reservoir, occlusion/blockage of needle/tubing, dislodged or bent needle, pump mechanical failure, low battery, detected over-infusion or under-infusion, patient adverse reaction (requiring immediate pump stop/alarm).	

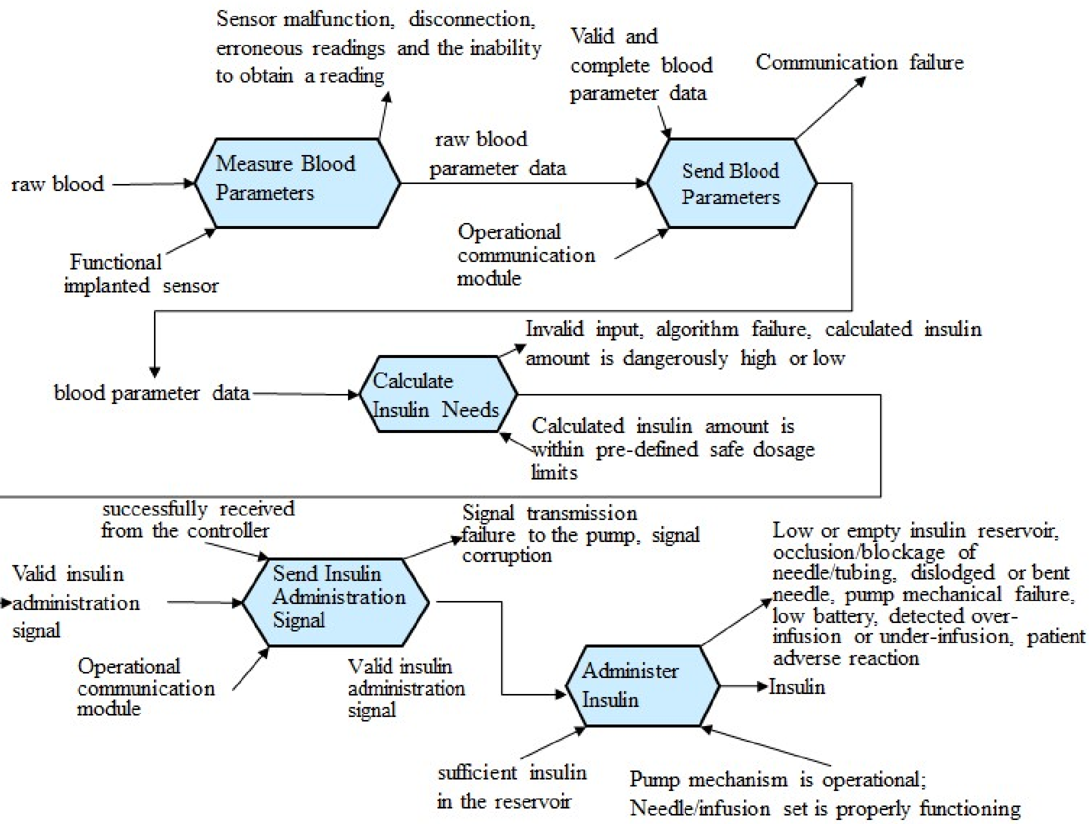

Figure 2. Defect Prevention Diagram of Insulin Pump Control System

Figure 2 shows the Defect Prevention Diagram based on the result of the Defect Prevention Matrix Analysis for the Insulin Pump Control System requirements.

C. PDCA

The following PDCA process description is then applied to the Prompt template DPMA.

Process Description: PDCA

Plan: Set goals and plan ways to achieve them.

Do: Take action based on the plan.

Check: Check the results and assess whether they went according to plan.

Act: Take improvement measures based on the evaluation and incorporate them into the next plan.

Generative AI developed the Defect Prevention Matrix as shown in Table VIII. Then the check results for the plan, do, check, and act process are shown in Tables IX to XI.

The DPM of PDCA clarifies the inter-relationship among the four processes of PDCA. The Table indicates that non-adjacent processes also have dependencies in the PDCA cycle. For example, there are three backward loops from the Do, Check, and Act processes to the Plan process. In addition, the PDCA cycle usually only shows the loop from Act to Plan, but the Table shows that Plan also has a relationship with Check and Act.

TABLE VIII. DEFECT PREVENTION MATRIX OF PDCA

M(X, Y)	Plan (P)	Do (D)	Check (C)	Act (A)
Plan (P)	Purpose: Set goals and plan ways to achieve them.	Provides input and resource conditions for Do.	Sets judgment criteria for Check.	Provides baseline for improvement in Act.
Do (D)	Uses plan to execute actions.	Purpose: Take action based on the plan.	Supplies results for evaluation in Check.	Provides operational data for Act.
Check (C)	Uses Do's output to evaluate performance.	Assesses execution against Plan.	Purpose: Check results and assess alignment with plan.	Provides evaluation results for Act.
Act (A)	Uses Check's evaluation to improve future plans.	Refines execution strategy for Do.	Adjusts judgment criteria for Check.	Purpose: Take improvement measures and feed them into the next Plan cycle.

Tables IX to XII show the checklists of PDCA processes. The checklists clarify the successful conditions of the PDCA cycle. If these successful conditions are not clear, the PDCA cycle will result in unnecessary rework. However, these success criteria seem not to have been given much importance when the PDCA cycle was first introduced.

Defect prevention analysis clarifies the success conditions for the PDCA process, promoting an efficient PDCA cycle. It also enables quick responses to exceptions that occur in each PDCA process, reducing the number of iterations of the outer loop of PDCA.

Therefore, by integrating defect prevention analysis with the PDCA cycle, business improvement through PDCA can be made more efficient. Defect prevention matrix analysis can clearly define the relationships between the processes that make up the PDCA cycle, which may allow for the prediction of PDCA cycle iterations in advance.

TABLE IX. CHECKLIST OF THE PLAN PROCESS

Aspect	Issues
Input	Goals may be vague or not measurable. Lack of historical data or context.
Accepting condition	No formal criteria for initiating planning. Planning may begin without stakeholder alignment.
Resource Condition	Insufficient tools or data for effective planning. Lack of cross-functional input.
Judgement condition	No clear success metrics defined. Risk assessment may be missing.
Output	Plans may be too abstract or not actionable. No version control or documentation standards.
Exception condition	No fallback strategy if planning fails. Unclear escalation path for unresolved planning conflicts.

TABLE X. CHECKLIST OF DO PROCESS

Aspect	Issues
Input	Plans may be misinterpreted or incomplete. Lack of clarity in task assignments.
Accepting condition	No readiness check before execution. Missing confirmation of resource availability.
Resource Condition	Tools or personnel may be underprepared. No contingency resources.
Judgement condition	Execution decisions may lack flexibility. No real-time monitoring or feedback.
Output	Results may not be recorded systematically. No linkage to performance indicators.
Exception condition	No protocol for handling execution failures. Delays may not be communicated upstream.

TABLE XI. CHECKLIST OF CHECK PROCESS

Aspect	Issues
Input	Data from Do may be incomplete or inconsistent. No baseline for comparison
Accepting condition	No criteria for when to initiate evaluation. Evaluation may be skipped under time pressure.
Resource Condition	Lack of analytical tools or expertise. No access to historical benchmarks.
Judgement condition	Evaluation may be biased or subjective. No structured review process.
Output	Findings may not be actionable. No feedback loop to Plan or Act.
Exception condition	No handling of conflicting evaluation results. Errors in data may go unnoticed.

TABLE XII. CHECKLIST OF ACT PROCESS

Aspect	Issues
Input	Evaluation results may be unclear or delayed. No prioritization of improvement actions.
Accepting condition	No criteria for implementing changes. Resistance to change may block action.
Resource Condition	Lack of authority or resources to implement changes. No support for training or adaptation.
Judgement condition	Decisions may be reactive rather than strategic. No long-term impact analysis.
Output	Improvements may not be documented or tracked. No integration into future planning.
Exception condition	No plan for failed improvement attempts. Feedback may not reach the right stakeholders.

V. DISCUSSION

A. *Novelty*

In this paper, we have proposed a procedure for creating the defect prevention analysis matrix and a review method, as well as an automatic execution procedure using generative AI.

1) *Structuring Business Process Knowledge*

In a defect prevention diagram, business process knowledge can be organized hierarchically using L1: business process knowledge, L2: business flow-related knowledge, L3: business process action conditional knowledge, and L4: business action condition execution knowledge. Here, L1, L2, and L3 can be described in a defect prevention diagram. However, for L4, the described conditions must be evaluated when the actual business process is executed.

In the business process, knowledge of a defect prevention diagram, L1 can grasp the overall picture of the business process by identifying the necessary actions that make up the business process. Business flow-related knowledge L2 can recognize the dependencies between business processes. Business process action condition knowledge L3 can recognize the necessary conditions to execute a business. The difference between L3 and L4 is the difference between knowing the conditions and being able to appropriately confirm and evaluate those conditions. Condition evaluation knowledge L4 should be specified so that the evaluation results do not vary depending on the individual for the same conditions.

In the defect prevention diagram, this type of business process knowledge classification is used to organize business knowledge that has traditionally been thought to vary between individuals, making it possible to clarify where these variations in knowledge occur.

The defect prevention diagram was originally proposed as the name of the business process completeness diagram [20].

2) Automating defect review with generative AI

As shown in the application example, the proposed Prompt template easily automates the review of business processes using the defect prevention matrix, demonstrating the applicability of the proposed method.

In the defect prevention diagram shown in Figure 3, created by humans [16], the details of the exceptions were insufficient. In contrast, Tables III to VI extracted by the generative AI show that specific exception candidates were elicited.

B. *Applicability*

In this paper, we confirmed the applicability of the proposed method by applying it to the train operation monitoring process, the PDCA cycle, and the insulin pump control system. Because these cases are important business processes for safety operation monitoring, the proposed method may apply to a wider range of safety-critical processes.

Furthermore, as we can see from Figure 2, the defect prevention diagram of the insulin pump control system only has linear relationships between the preceding and following processes, whereas the process relationships of the Shinkansen bogie inspection system and the PDCA cycle have comprehensive interactive relationships between the processes.

C. *Comparison with RCA and FRAM*

In Root Cause Analysis (RCA), when a defect is detected in a system, the cause of the defect is identified. Once the cause is identified, measures are devised to prevent the defect from occurring in the system. In contrast, in defect prevention analysis, which is the premise of the defect prevention diagram, the success conditions and exceptions of the system are first defined. Next, measures to deal with exceptions are devised in the system. Defects that occur during the operation of the system are identified, and the planned measures are implemented.

FRAM and the Defect Prevention Matrix Analysis (DPMA) have common aspects as input, output, and resources. FRAM has time, precondition, and control aspects that are not in DPMA. DPMA also has acceptance, exception, and judgment condition aspects, which are not in FRAM. The output of FRAM is restricted to output aspects. Therefore, the meaning of output in FRAM may be unclear as it is difficult to discriminate exceptional output from normal output by aspects.

Although there are differences between FRAM and DPMA, it is unclear whether they have the same expressive power. As FRAM can be applied to analyze the resonance relationship between processes, the completeness of

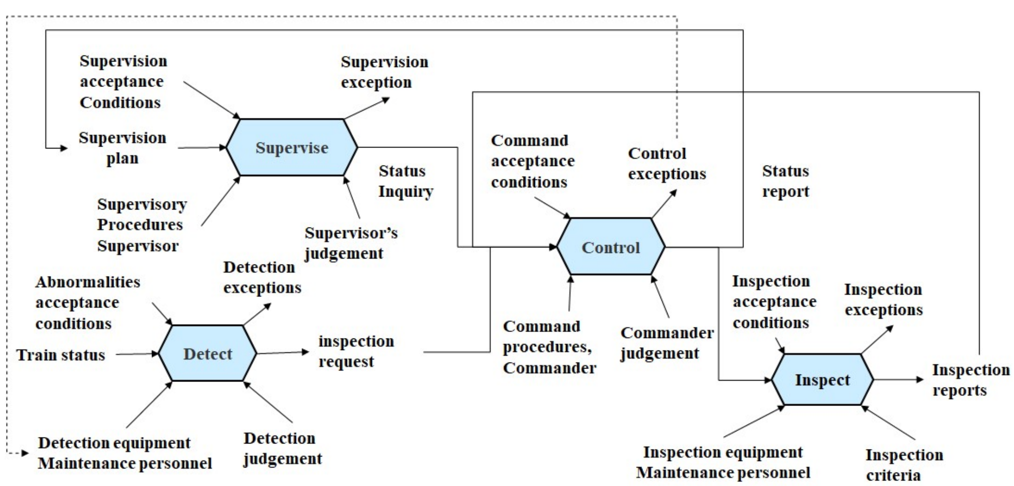

Figure 3. Defect Prevention Diagram of Shinkansen Bogie Inspection System

business processes may also be possible to analyzed by FRAM. Sujan and Felici [21] combine Failure Mode and FRAM. This implies a new method possibility that integrates the analysis method using DPMA with Failure mode analysis. FRAM attempts to model the diversity and interactions of a system in detail, but if the model is too detailed, it becomes overly complex, making analysis difficult and impractical insights difficult to obtain. It's difficult to determine how much diversity to describe and at what granularity to define functions, and excessive modeling can increase analysis time and cause essential insights to be overlooked.

Even if the mechanism of variation can be identified, specific countermeasures may not be obvious. Further consideration is required to identify the most effective interventions among the complex interactions.

FRAM is not simply a template-based approach; it requires a skilled facilitator who can deeply understand and facilitate discussion about the functions that make up the system, their interactions, and the sources of diversity. Attempting to apply FRAM without experience can result in a superficial analysis or overlooking important aspects, resulting in ineffective results.

The advantages of DPMA over FARM are as follows:

1) DPMA can derive success conditions based on the business process definition. FARM requires identifying functions from the business process, which not only leaves the granularity of the function definition indefinite, but also makes the optimal granularity dependent on personal judgment. Furthermore, FRAM lacks clear standards for extracting success conditions, making consensus-building among stakeholders essential.

2) FRAM explores success and failure conditions based on functional resonance, assuming that the execution conditions of functions are unclear, making it difficult to ensure the integrity of business processes. DPMA clearly defines process integrity conditions, allowing deviations from those conditions to be detected and addressed as exceptions. Without defining normal conditions, it is impossible to guarantee the safety of business processes.

The formal comparison between FRAM and DPMA is an interesting future research theme.

Table XIII shows the comparison of RCA and FRAM to DPMA.

TABLE XIII. COMPARISON WITH RCA AND FRAM

	RCA	FRAM	DPMA
Overview	Investigate the cause of the problem	Success-based integration	Comprehensive Defect Prevention
Direction	Backward analysis	Upstream/downstream analysis	Forward/ Backward Analysis
Implementation Period	Post-incident	Pre- and post-event analysis	Proactive
Countermeasures	Prevent recurrence	Prevention of recurrence	Prevention
Focus Points	Occurring problem	Functional fluctuation	Success Conditions, Exceptions, and Responses
Methods	Cause-and-effect analysis	Functional resonance analysis	Exception Propagation Analysis
Challenges	Expand to other problems	Identification and comprehensiveness of aspects and resonance relationships	Overall Optimization and Residual Risk

D. *Exception propagation*

We need to clarify how to respond to process exceptions. In the case study, we identified exception detection but not propagation. We can define a template to perform the exception propagation procedure as shown below. The template that performs the exception propagation procedure is as follows.

Prompt template: Exception propagation.

Identify the exception propagation relationships between processes for <the Processes>.
The exception propagation procedure is as follows:
The exceptional propagation relationships between processes are as follows:
Propagation to the receiving condition of the preceding process. Propagation to the decision condition of the preceding process. Propagation to the resource condition of the current process. Propagation to the receiving condition of the subsequent process. Propagation to the decision condition of the subsequent process

The template is applied to the PDCA process. The results of applying for the PDCA process are shown below.

TABLE XIV. Exception Propagation of PDCA cycle

Process	To	Exception	Impact
Plan	Resource Condition of Current Process	Lack of data, unclear goals, or unavailable planning tools	Planning cannot proceed effectively
Plan	Receiving Condition of Subsequent Process (Do)	Ambiguous success criteria	Execution may be delayed or misaligned
Plan	Decision Condition of Subsequent Process (Do)	Ambiguous success criteria.	Execution decisions may lack clarity or direction.
Do	Receiving Condition of Preceding Process (Plan)	Execution reveals flaws in the plan.	May require re-planning or plan revision.
Do	Decision Condition of Preceding Process (Plan)	Execution deviates from expected outcomes.	Planning assumptions may be invalidated.
Do	Resource Condition of Current Process	Lack of tools, personnel, or time.	Execution is compromised or halted.
Do	Receiving Condition of Subsequent Process (Check)	Poor documentation or missing data.	Evaluation becomes difficult or inaccurate.
Do	Decision Condition of Subsequent Process (Check)	Inconsistent results.	Evaluation may be inconclusive or misleading.
Check	Receiving Condition of Preceding Process (Do)	Evaluation identifies critical execution errors.	May require re-execution or correction.
Check	Receiving Condition of Preceding Process (Do)	Results contradict expected outcomes.	Execution decisions may be questioned.
Check	Resource Condition of Current Process	Lack of analytical tools or expertise.	Evaluation is incomplete or biased.

Check	Receiving Condition of Subsequent Process (Act)	Evaluation results are unclear or delayed.	Improvement actions may be postponed.
Check	Receiving Condition of Subsequent Process (Act)	Conflicting evaluation findings.	Decisions on improvements may be uncertain.
Act	Receiving Condition of Preceding Process (Check)	Improvement actions contradict evaluation.	May require re-evaluation.
Act	Decision Condition of Preceding Process (Check)	Changes invalidate previous assessments.	Evaluation criteria may need revision.
Act	Resource Condition of Current Process	Lack of authority, tools, or support for change.	Improvements are not implemented.
Act	Receiving Condition of Subsequent Process (Plan)	Feedback not incorporated into new plans.	Planning cycle does not evolve.
Act	Decision Condition of Subsequent Process (Plan)	Lessons learned not reflected in planning criteria.	Future plans repeat past mistakes.

The exception propagation for the Shinkansen bogie inspection system created by humans in Figure 3 does not specify exceptions and only extracts the exception propagation from the Control process to the Resource condition of the Defect process. In contrast, as shown in Tables III to VI, generative AI comprehensively extracts exceptions. Furthermore, in the exception propagation for the PDCA cycle, it extracts exception propagation not only to preceding and succeeding processes but also to the process itself. Therefore, exception propagation analysis by generative AI is more comprehensive than exception propagation created by humans.

Conversely, extracting comprehensive exception propagation has the problem of complicating the defect prevention diagram. Since it is not known whether all exception propagations will be necessary, the importance of exception propagation must be evaluated. In the future, a mechanism for evaluating the priority of exception propagation will need to be devised.

Another option is to stop all business processes when an exception occurs in order to respond to the exception. In terms of business process continuity, it is necessary to compare the method of continuing the execution of partial processes using exception propagation with the method of stopping all processes and responding to the exception. In the future, it will be necessary to consider criteria for evaluating the effectiveness of both methods.

E. *Sufficiency of Six Process Aspects*

The purpose of this paper is not to analyze complex system interactions, but to analyze defects in business processes. We propose extending the traditional self-process-containment method with exceptions and automating the procedure using generative AI. We do not claim that defects can be expressed using only six aspects. However, actual business processes can be expressed using the proposed five aspects and their exceptional aspects. Toyota has refined its manufacturing process to produce high-quality automobiles using a self-process-containment process. This demonstrates the practical sufficiency of the six aspects. Anyone with experience of designing actual business processes should be able to understand the sufficiency of the six aspects. Furthermore, our application to actual accident cases demonstrates that this is a practical defect analysis method.

The reasons why PRC based on the six aspects is not incomplete are as follows: First, it is always true whether the conditions of the five aspects are met. Second, an exceptional condition occurs when the five aspects are not met. Therefore, PRC based on the six aspects is sufficient.

F. *Comparison with Learning Methods*

The purpose of this paper is not to compare machine learning methods, but to demonstrate the possibility of automating the newly proposed business process defect analysis method using generative AI. Therefore, alternative methods for defect detection (such as supervised/unsupervised machine learning or GNNs) cannot replace the Defect Prevention Matrix. Generative AI is used as a means to automate defect analysis of process design in production and business processes. Copilot is used as the generative AI in this example. Similar results can also be obtained with ChatGPT and Gemini.

Comparison with business process defect detection methods is also made with methods such as FRAM and FMEA.

Furthermore, machine learning approaches require training data, which incurs unnecessary costs for use in business settings. Machine learning has strengths in detecting "visual defects" and "physical anomalies," but the same techniques cannot be applied to business process defects because they are non-visual, context-dependent, and dynamically defined.

In contrast, generative AI can execute procedures, eliminating redundant training costs. As described above, the purpose of this paper is to demonstrate the feasibility of using generative AI to automate defect analysis of maintenance and operation processes. Comparing the capabilities of generative AI and machine learning methods is beyond the scope of this paper and will be addressed in the future.

G. *Limitation*

In this paper, we have proposed a method for reviewing defect prevention diagrams. However, we have only applied it to a few cases. In the future, we need to quantitatively evaluate the effectiveness of the method by applying it to many cases.

However, because the number of exceptions detected by generative AI and their propagation will be large, we plan to continue studying how to select and discard them based on the importance of the detection results. We also need to clarify the criteria for determining whether to stop a business process when an exception is detected or to partially execute it based on the exception propagation response. Furthermore, since this paper only conducted a desk-based trial evaluation, it is necessary to apply it to more case studies and actual business processes for evaluation.

VI. Conclusion

This paper proposes a review method using a defect prevention diagram, a process relationship table, and a checklist. The business process review checklist can verify the completeness of each of the six aspects of the processes that make up a defect prevention diagram. In particular, it can detect inconsistencies between multiple inputs and outputs. Furthermore, the process relationship matrix can analyze the comprehensive dependencies between the business processes that make up a defect prevention diagram.

By defining transition relationships based on the elements of the business relationship matrix M, it is possible to track influence relationships iteratively. In other words, M can be used to define a linguistic expression L for a defect prevention diagram. It is believed that this L can be used to formulate the equivalence of defect prevention diagrams, thereby minimizing the size of the defect prevention diagram.

The defect prevention diagram can complement the response to exceptions in business processes, making it possible to define business processes that can handle defects as exceptions.

This paper also clarifies prompt templates for a generative AI to automate the proposed method. Furthermore, the effectiveness of the proposed method was confirmed by applying it to a Shinkansen bogie inspection system, an insulin pump control system, and a PDCA cycle.

In this paper, we formulated the completeness of defect prevention diagrams in terms of their ability to respond to exceptions. We also demonstrated that comprehensive verification is possible through the automation of exception detection and exception propagation using generative AI.

However, because the number of exceptions detected by generative AI and their propagation will be large, we plan to continue studying how to select and discard them based on the importance of the detection results. We also need to clarify the criteria for determining whether to stop a business process when an exception is detected or to partially execute it based on the exception propagation response. Furthermore, since this paper only conducted a desk-based trial evaluation, it is necessary to apply it to more case studies and actual business processes for evaluation.

References

[1] S. Yamamoto, “Defect Prevention Review by Process Relationship Matrix” The Seventeenth International Conference on Information, Process, and Knowledge Management (eKNOW 2025) IARIA, 2025, pp. 37-40.

[2] S. Sasaki, “Self-process completion - Quality is built in the process,” JSQC selection, Japan Society for Quality Control, 2014. (in Japanese)

[3] J. Salvadorinhoa and L. Teixeira, “Organizational knowledge in the I4.0 using BPMN: a case study,” CENTERIS 2021, Procedia Computer Science 181, 2021, pp. 981–988.

[4] A. Ore, O. Kuznecova, and A. Jegorova, “Self-managed Organization: A Role of Business Process Management,” 2021 62nd International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS), doi: 10.1109/ITMS52826.2021.9615260

[5] P. Faraboschi, E. Frachetenberg, P. Laplante, D. Milojicic, and R. Saracco, “Digital Transformation: Lights and Shadows,” IEEE Computer, 2023, pp. 123-130.

[6] S. Yamamoto, “A Strategic Map for Digital Transformation,” KES 2020, Procedia Computer Science, vol. 176, 2020, pp. 1374-1381.

[7] S. Yamamoto, “A Comparative Analysis of Business Model Notations,” Journal of Business Theory and Practice, ISSN 2372-9759 (Print) ISSN 2329-2644 (Online), vol. 7, no. 3, 2019.

[8] D. Leonard and W. Swap, “Deep Smarts: How to Cultivate and Transfer Enduring Business Wisdom,” Harvard Business Review Press, 2005.

[9] Z. Nakamura, “How to conceive a simple job—Succeed in Entity Process Analysis,” Nikkan Kogyo Shimbun, 2003 (in Japanese)

[10] S. Yamamoto and H. Fujimoto, “PKC-- Production Knowledge Chart for Knowledge Transfer,” KES2023, Procedia Computer Science 225, 2023, pp. 414-423.

[11] D. Dori, “Model-Based Systems Engineering with OPM and SysML,” Springer, 2016.

[12] B. Cameron, E. Crawly, and D. Selva, “System Architecture,” PEARSON, 2015.

[13] E. Hollnagel, “FRAM - the Functional Resonance Analysis Method: Modelling Complex Socio-Technical Systems.” CRC Press, 2012.

[14] J. Lundberg, R. Woltjer, “The Resilience Analysis Matrix (RAM): Visualizing functional dependencies in complex socio-technical systems,” In: Proceedings of the 5th Resilience Engineering Association Symposium, 2013.

[15] R. Patriarca, G. Pinto, G. Di Gravio, F. Costantino, “FRAM for Systemic Accident Analysis: A Matrix Representation of Functional Resonance,” International Journal of Reliability Quality and Safety Engineering, vol. 25, no. 1, 2018, doi: 10.1142/S0218539318500018

[16] S. Yamamoto, “Comparative Study on Functional Resonance Matrices,” Knowledge-Based Software Engineering: 2022: Proceedings of the 14th International Joint Conference on Knowledge-Based Software Engineering (JCKBSE 2022), M. Virvou, T. Saruwatari, L. Jain *Editors,* Learning and Analytics in Intelligent Systems 30, Springer, 2023, pp. 169-180, doi:org/10.1007/978-3-031-17583-1

[17] dataspan.ai blog, Automated Defect Detection: How GenAI and Synthetic Data are Transforming Visual Inspection in Manufacturing, July 2025, https://www.dataspan.ai/blog/automated-defect-detection-how-genai-and-synthetic-data-are-transforming-visual-inspection-in-manufacturing.

[18] A. R. Singh and others, A comprehensive survey of image synthesis approaches for Deep Learning-based surface defect detection in manufacturing, Computers in Industry, ScienceDirect Elsivier, Computers in Industry, Volume 173, ISSN 0166-3615, https://doi.org/10.1016/j.compind.2025.104360.2025.

[19] I. Sommerville, Software Engineering Tenth Edition, Addison-Wesley, 2015.

[20] S. Yamamoto, “Business Process Completeness” The Sixteenth International Conference on Information, Process, and Knowledge Management (eKNOW 2024) IARIA, 2024, pp. 24-28.

[21] M. Sujan and M. Felici, “Combining Failure Mode and Functional Resonance Analyses in Healthcare Settings,” SAFECOMP, LNCS 7612, 2012, pp. 364-375.