

# Correct and Control Complex IoT Systems: Evaluation of a Classification for System Anomalies

Sina Niedermaier  
*Institute of Software Technology*  
*University of Stuttgart*  
 Stuttgart, Germany  
 sina.niedermaier@iste.uni-stuttgart.de

Stefan Heisse  
*Bosch Engineering GmbH,*  
 Stuttgart, Germany  
 stefan.heisse@de.bosch.com

Stefan Wagner  
*Institute of Software Technology*  
*University of Stuttgart*  
 Stuttgart, Germany  
 stefan.wagner@iste.uni-stuttgart.de

**Abstract**—In practice there are deficiencies in precise inter-team communications about system anomalies to perform troubleshooting and postmortem analysis along different teams operating complex IoT systems. We evaluate the quality in use of an adaptation of IEEE Std. 1044-2009 with the objective to differentiate the handling of fault detection and fault reaction from handling of defect and its options for defect correction. We extended the scope of IEEE Std. 1044-2009 from anomalies related to software only to anomalies related to complex IoT systems. To evaluate the quality in use of our classification a study was conducted at Robert Bosch GmbH. We applied our adaptation to a postmortem analysis of an IoT solution and evaluated the quality in use by conducting interviews with three stakeholders. Our adaptation was effectively applied and inter-team communications as well as iterative and inductive learning for product improvement were enhanced.

**Index Terms**—anomaly, failure, fault, defect, complex system, classification

## I. INTRODUCTION

Providers of cloud-based IoT solutions compile distributed complex systems of components, integrating hardware, software and mechanical system elements. In operations they aim for controlling their dynamic behaviour in a permanently changing context of use. The system components are developed and operated by different teams from different organizational units. The teams are often specialized in the context of their component and their organizational silo. Communication within an organizational silo and the teams is performed by applying concepts that are optimized for their specific intra-team context.

In case of failure, a failed function of an IoT service, the different teams operating the system components, have to form a collective to collaboratively control the situation.

First, in situation management (troubleshooting), facing dynamics, the objective is to regain control as quickly as possible with fault reactions. The purpose of fault reaction is to avoid a failure influencing customer satisfaction or, if not successful, at least to provide limited operation for keeping the customer's trust by controlling the situation. Second, once the system is transitioned back into a controlled state, the anomaly in the effect chain and the actions of responding to dynamic cascades of consecutive anomalies are investigated within a postmortem analysis. The objective is to detect defects and

subsequently remove them.

Each of the two sets of activities, situation management and postmortem analysis follow different optimization goals. First, detection of service failure initiates tactical actions performed with the goal of customer satisfaction. The leading principle is to 'be effective before efficient'. Second, the subsequent analysis to identify defects and options for correction initiates strategical actions performed with the goal of profitability. The leading principle is to 'be efficient'.

To perform situation management and postmortem analysis several development and operational teams from different organizational silos form a collective. Deficiencies on shared concepts for communication about anomalies often lead to misunderstandings and different interpretations regarding anomalies. Missing shared concepts on anomalies for inter-team communications across organizational silos can be used to direct "responsibility" for failure, its compensation and related defect correction to a specific organizational unit [1].

Assigning blame to a specific organizational unit can be intensified by performing root cause analysis, for example by asking 5xWhy [2]. While doing this, people tend to search for isolated individual causes. In retrospective, humans strive to oversimplify causal chains to a single root cause without embracing complexity and dynamics in cause effect chains [3].

As stated by Cook [4] for distributed complex systems there might be no isolated root cause. Rather, there are multiple faults and contributors, where each fault is necessary but they are only jointly sufficient for failure [4]. Asking "why" tends to point in the direction of "who". This oversimplification to a single root cause and a single responsibility often results in a culture of blaming. In contrast, a retrospective is an instrument of a learning organization performing feedback cycles to improve product and service quality as well as organization quality [5]–[7]. The feedback cycles process defects related to product and service quality and defects related to the performance of the organization.

If the differentiation of effective tactical work mode from efficient strategic work mode is missing, an organization tends to remain in effective tactical mode. If a failure can be controlled only by tactical fault reactions, an interest in the defect and its sustainable correction often remains secondary. Therefore, defects with their options for correction are not

listed and accumulate to technical debts.

To address this need for a precise differentiation and to enhance the effectiveness and efficiency of the collaboration capabilities of developers, operators and organizations, we propose a shared classification for anomalies. This work is based on the common logical concept of the IEEE Std. 1044-2009 [8] to classify software anomalies.

We propose an adaptation of the IEEE Std. 1044-2009 applied only to system element software to extend the same for application to a system composed of a set of interacting system elements. This enables differentiation between the handling of fault detection and dynamic fault reactions from the handling of defect and its options for sustainable defect correction, not only on system element software but generally on system level.

We investigated its quality in use [9] by conducting a case study with two stages. In the first stage we applied our adaptation of the IEEE Std. 1044-2009 to a postmortem analysis and present the results of the classification. To further evaluate the quality in use of our adaptation, we conducted semi-structured interviews with three of the stakeholders of the postmortem analysis.

## II. RELATED WORK

According to Wagner [10] there are several anomaly and defect classifications with different purposes, e.g. to enhance the identification of defects or the education of developers. One of them is the IEEE standard 1044-2009 [8] classifying software anomalies. The standard introduces the different concepts of problem, failure, fault and defect. The IEEE Std. 1044-2009 is the basis for our classification, therefore we introduce it in detail in Section III-A. Another widespread classification for anomalies is the taxonomy of dependable and secure computing by Avizienis et al. [11]. The authors address the threats to dependability and security by the concepts of failure, error and fault. The classification of this work has several symmetries to the classification of Avizienis et al. It is possible to roughly map their concepts of failure, error and fault to the concepts of failure, fault and defect of this work. The main difference of the concept of defect of this work to their concept of fault is that they consider the fault in a static as well as in a dynamic system view and do not differentiate between them.

With the concept of postmortems, critical service disruptions are analyzed in retrospective with a focus on how well the organization responded and recovered from the disruption. Postmortems are an essential part of Google's discipline of Site Reliability Engineering [12], that is increasingly being used in various companies [1]. Part of the approach of postmortems is the concept of root cause analysis (RCA) [12]. RCA aims to identify the root of an anomaly and also why it was introduced. Thereby, RCA targets to prevent similar causes from being introduced in the future [10]. As we refer to the work of Cook [4], who asserts that there is no root cause in complex system, with this work, we try to avoid the concept of searching for single causing entity, but rather to identify defects with their options for correction.

## III. CONCEPTS

### A. Classification for Anomalies

The IEEE Std. 1044-2009 is a standard which provides a unified approach to classify software anomalies and models the relation between software anomalies and maintenance activities [8]. Our adaptation of the concepts of the IEEE Std. 1044-2009 is general enough to cover not only software but also anomalies related to complex IoT systems. An (IoT) system in this work is defined by its components, compiled into a structural arrangement interacting in component-effect relationships. The components itself are composed of interacting system elements such as software, hardware, mechanics, humans, etc. [13].

We present the concepts of software anomalies of IEEE Std. 1044-2009 [8] and differentiate our concepts applicable to complex systems providing IoT services.

- **Problem:** Difficulty or uncertainty experienced by one or more persons, resulting from an unsatisfactory encounter with a system in use.
- **Failure:** An event in which a system or system component does not perform a required function within specified limits. (adapted from ISO/IEC 24765:2009 [14])
- **Fault:** A manifestation of an error in software. (adapted from ISO/IEC 24765:2009 [14])
- **Defect:** An imperfection or deficiency in a work product where that work product does not meet its requirements or specifications and needs to be either repaired or replaced. (adapted from the Project Management Institute [15])

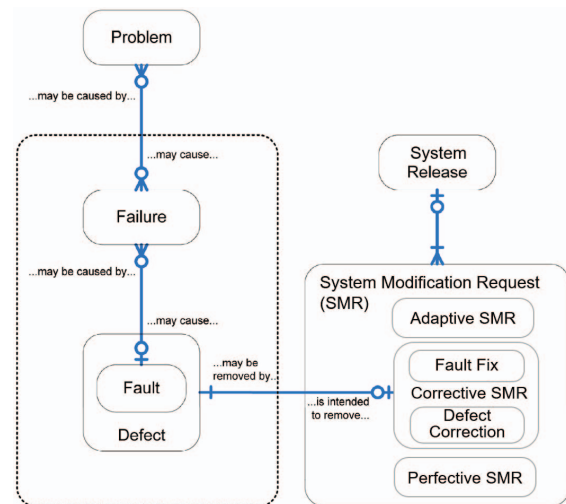


Fig. 1. Relationships modeled as an entity relationship diagram (adapted from [8])

The relationships between the different concepts are shown in Figure 1. Further the IEEE Std. 1044-2009 [8] depicts the relationship between the concepts **fault** and **defect**, where: “A fault is a subtype of the supertype defect. Every fault is a defect, but not every defect is a fault. A defect is a fault if it is encountered during software execution (thus causing a

failure). A defect is not a fault if it is detected by inspection or static analysis and removed prior to executing the software.”

For developing and operating complex IoT systems three architectural views are motivated. “A view is a representation of a whole system from the perspective of a related set of concerns” [16].

- **System structure view:** The system structure view presents the set of components in their structural arrangement. The system structure view is optimized to organize the division of labour.
- **Functional system view:** The functional view breaks down the customer function into sub-functions that can be assigned to the system components. The functional system view is optimized to represent the value proposition to the customer.
- **Dynamic system view:** The dynamic view represents the component effect<sup>1</sup> relationships in temporal expanse. The dynamic system view is optimized for situation management to control the dynamic behaviour.

To avoid conflicts with other concepts of anomalies, we choose a deductive approach and motivate the concepts of anomaly out of the system views. The views contain conformities and nonconformities. We define a nonconformity related to a system view as an anomaly. The anomalies of IEEE Std. 1044-2009 can be mapped to the anomalies of the three system views.

- **Failure:** A failure is a failed function, where a system does not perform a required function within specified limits. This concept of anomaly is assigned to and optimized for the **functional system view**.
- **Fault:** A fault is a break in a component effect chain and has a temporal expanse. This concept of anomaly is assigned to and optimized for the **dynamic system view**. A fault can be classified as permanent, transient or intermittent [17].
- **Defect:** Whereas the defect in IEEE Std. 1044-2009 is an imperfection or deficiency in a work product, we assign the defect to a subset of components in their structural arrangement and identify *potential option(s) for correction*. Option(s) for correction are related to modification on a:
  - subset of components
  - component itself
  - system element of a component

The trivial option for correction is the exchange of the subset of components the defect is assigned to. An assignment of option(s) for correction to a subset of components does not imply poor quality of work. The concept of anomaly defect and its option(s) for correction are assigned to and optimized for the **system structure view**.

<sup>1</sup>For this work, component effect relationship is equivalent to cause effect relationship. A cause is related to a component, therefore we use the concept of component effect chain.

We adapted the software change request (IEEE 1044-2009) to a system modification request (SMR)<sup>2</sup> [18] assigned to a subset of components in structural arrangement.

As a fault is a defect, there is a (see Figure 1):

- **1<sup>st</sup>** corrective SMR resulting in a (quick) **fault fix** and a
- **2<sup>nd</sup>** corrective SMR resulting in a **defect correction**.

### B. Fault Tolerance and Situation Management

Within this section, we focus on the relationship between the concept fault and concept failure. In engineering complex IoT systems we aim for fault tolerance: “the ability to deliver a specified functionality in the presence of one or more specified faults” [17]. Ideally, a system is designed fault tolerant. If the system is not tolerant to a fault, we face a situation: we have to react to keep customer’s trust. Now time matters. We are bound to time and switch into effective mode.

Analogous to ISO 26262 – Functional safety – applied in engineering control systems [17], we differentiate the following time stamps and time intervals in the **dynamic system view** (see Figure 2).

Time stamps:

- $t_1$ : Occurrence of fault, fault is not detected.
- $t_2$ : Time when fault is detected.
- $t_3$ : End of fault reaction.
- $t_4$ : End of fault tolerant time interval. Occurrence of failure.

Time intervals:

- **FTTI** (Fault tolerant time interval) is the minimum time span from the occurrence of a fault in a component-effect chain of a system (dynamic view) to a possible occurrence of a failure if the quality mechanism is not activated.
- **FDTI** (Fault detection time interval) is the time span from the occurrence of a fault to its detection.
- **FRTI** (Fault reaction time interval) is the time span from the detection of a fault executing a fault reaction until a transition to a controlled state is achieved.

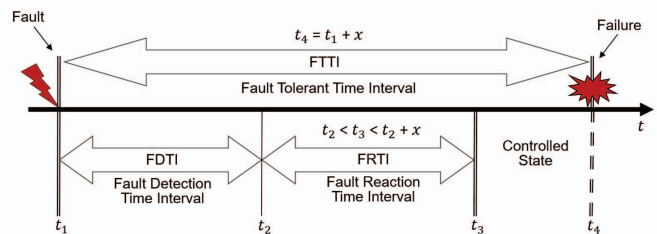


Fig. 2. Fault tolerant time interval (FTTI) (adapted from [17])

**Fault Tolerance:** For improving fault tolerance we automate fault reactions. Fault reaction is performed to prevent the fault from progressing to failure. The fault is detected in FDTI.

<sup>2</sup>For this work, the system modification request includes the software change request. A SMR in virtual systems focuses on a re-compiled (corrected) structural re-arrangement of a subset of components and not on modifying system elements of components.

Fault reaction is performed in FRTI and has to be successful before FTTI ends.

**Situation management:** If the system is not tolerant to a fault, detecting the fault we switch into a tactical mode by performing an OODA loop [19]. We **Observe, Orient, Decide** and **Act** (OODA) in the dynamic system view. For keeping the customer's trust we demonstrate that we control the situation. In FTTI, we transfer the system into a controlled state that is predefined and agreed for customer context of use. We engineer in having tactical options: we prepare quality mechanisms, fault reactions which transfer the system into a controlled state. Controlled states include:

- **redundancy activated:** no failure in customer context [11].
- **degraded:** permission to use a service that does not conform to specified requirements [7] e.g. graceful degradation by serving stale data [20].
- **regraded:** alteration of the grade of a nonconforming service in order to make it conform to requirements differing from the initial requirements (related to value proposition) [7].
- **terminated:** preclude usage by controlled take out of service [7] e.g. isolation. [11]
- **compensated:** compensation of damages for the customer, e.g. in the form of monetary award.

In situation management the customer is watching us and assesses behaviour and communications and decides to continue placing trust on us. Collaborating with the customer, precise communication about anomalies relies on shared concepts.

If the (quick) fault fix is effective, the system is in a controlled state. We control the situation. The customer trust remains placed on us. Now we are independent from time, we leave the dynamic system view. We switch into efficient mode. We perform quality management, in particular defect management in a PDCA loop (**Plan, Do, Check, Act**) [7]. During a postmortem analysis, improvement for fault reaction to increase fault tolerance as well as improvement for tactical options to increase capability for situation management can be identified and transferred to defect management.

### C. Defect Correction and Quality Management (Defect Management)

Sustainable defect correction, which is optimized for strategic interest is not equivalent to a (quick) fault fix, which is optimized for tactical interest. Now, we act in efficient mode with the goal of profitability.

In postmortem analysis, we identify anomalies related to the different system views:

- In the functional system view an anomaly (failure) with need for correction is identified, a defect is detected and is to be assigned to system structure view.
- In the dynamic system view an anomaly (fault) with the need for correction is identified, a defect is detected and is to be assigned to system structure view.

- In the system structure view an anomaly (defect) with the need for correction is detected and is to be assigned to a subset of components.

As described in Section III-A, a defect and its option(s) for correction are assigned to **system structure view** and a subset of components in their structural arrangement. The defect management decides whether and which option for defect correction, addressed by a SMR, is to be executed for strategic and efficiency interest.

The lifecycle of a defect is depicted in Figure 3.



Fig. 3. Defect life cycle (adapted from [8])

We adapted the UML statechart diagram of the defect life cycle of the IEEE Std. 1044-2009 [8]. The different states of a defect *inserted*, *detected*, *removed* have been extended with an additional state *graced*. We propose, that after a fault is detected, there is the additional option to grace a defect for a certain period of time until it is removed. The decision to grace a defect is based on an assessment that a defect of a complex IoT system is accepted under following circumstances:

- The decision can take place due to economical reasons, when the organization decides to remain in a fault reaction mode (as described in Section III-B) instead of sustainably remove the defect.
- The decision to grace a defect can be influenced by the assumption that there will be acceptable customer impact.
- The decision to grace a certain defect may be necessary in the context of a safety vs. security vs. privacy discussion. For example, a defect from security perspective – a security defect – may be graced for safety or privacy reasons.

Nevertheless, defects have to be documented in a defect list and are to be transferred to defect management.

## IV. SCOPE AND RESEARCH METHOD

To structure our research, we applied the case study research process as proposed by Runeson and Hoest [21].

### A. Research Goal

The goal of this study is to evaluate the quality in use (with limitation to the quality sub-characteristics: effectiveness, efficiency and satisfaction) of an adaptation of the Standard Classification for Software Anomalies, the IEEE 1044-2009, by applying it to a postmortem analysis of an IoT System.

As part of our adaptation, we have extended the scope of IEEE Std. 1044-2009 from anomalies related to software only to anomalies related to complex IoT systems providing service to a customer. We differentiate between the handling of:

- first, **fault** detection and **fault reaction** in the context of **situation management** from



- second, **defect** and its options for **defect correction** in the context of **defect management**.

This work is intended to enhance effectiveness and efficiency of the collaboration capabilities between developers, operators along organizational units by providing common logical concepts of anomalies. The concepts of anomalies enable precise inter-team communication to differentiate between actions for controlling the system and actions for its sustainable correction.

For evaluation of the quality in use with its quality sub-characteristics effectiveness, efficiency and satisfaction the following definitions are presented and interpreted for the context of this work. According to ISO/IEC 25022 [9] **quality in use** is the “degree to which a product or system can be used by specific users to meet their needs to achieve specific goals with effectiveness, efficiency, satisfaction, and freedom from risk in specific contexts of use”.

**Effectiveness:** “Accuracy and completeness with which users achieve specified goals” [22]. We evaluate the accuracy of differentiating the handling of fault detection and fault reaction from the handling of defect detection and its option(s) for defect correction, according to our adapted classification.

**Efficiency:** “Resources expended in relation to the accuracy and completeness with which users achieve goals” [22]. Resources include temporal and also mental effort performing the task. Since we performed qualitative research we did not track time consumption measurements. We therefore qualitatively evaluated the estimated efficiency of the interview participants in comparison to other postmortem analyses.

**Satisfaction:** “Degree to which user needs are satisfied when a product or system is used in a specified context of use” [9]. We evaluated this characteristic by asking the interview participants about their subjective opinions and attitude towards applying the classification.

### B. Case Context

We evaluate the quality in use of the classification for system anomalies in a case study conducted in 2019 at Robert Bosch GmbH, a German company providing IoT solutions.

The case study is based on a postmortem analysis related to a customer problem in the domain of IoT condition monitoring. We designed the case study in a two-staged procedure. In stage 1, we applied the adapted classification on a postmortem analysis of the IoT solution. In stage 2, we performed semi-structured interviews to evaluate the quality in use of our concept with the stakeholders of the postmortem analysis.

### C. Stage 1 Application - Data Collection and Analysis

In this stage specialists, representing the stakeholders interest formed a collective to perform postmortem analysis. For analysis, the method of 5xWhy was executed to assign root causes. Selected stakeholders of the collective applied our classification for anomalies to enable intra-collective communications. The data collection technique can be determined as first degree where two of the researchers were in direct relation to the case company. One researcher acted as coach to ensure

the application of our classification during the postmortem analysis. The second researcher took the role of an independent observer. For data triangulation reasons, we analysed data from multiple sources by collecting incident report data, notes from postmortem analysis meetings, observations and mails discussing the classification for anomalies. As the data is sensitive it is not provided with this work. By member checking, the results were revised by the stakeholders of the postmortem analysis and have been simplified for the purpose of this article in Section V-C.

### D. Stage 2 Quality in Use Interviews - Data Collection and Analysis

To further evaluate the quality in use of the proposed classification for anomalies, we performed semi-structured interviews with one participant of each stakeholder group of the postmortem analysis. The interviews aim to explore the individual experiences of the interviewees and to qualitatively evaluate the quality in use of our classification. All of the interviewees were actively involved during the postmortem analysis and are responsible for the resulting classification. The stakeholders include the product manager (S1) of the IoT solution, a quality manager of the cloud infrastructure provider (S2) and a systems analyst and quality expert (S3), who has coached the other stakeholders in applying the classification. All of the three interviewees are employees of the company.

To conduct the interviews, we created an interview guide [23] which is structured in general questions about classifications for anomalies, main blocks about the quality in use of our classifications and open questions about pros and cons as well as potential for optimization. We audio recorded the three face-to-face interviews of approximately 40 minutes. After transcribing, we sent the interviews to the participants for review. For evaluating the quality in use of the classification, we analysed the semi-structured interviews by performing Mayring’s approach of qualitative content analysis [24]. We performed a mixed approach of deductive and inductive coding by encoding the transcripts with the predefined quality in use criteria of Section IV-A and creating further categories for contents which could not be directly assigned to the existing categories. The transcripts were analyzed on a sentence level. The results of the interviews are described in Section V-E.

## V. CASE STUDY

This section provides an abstract description of the IoT solution and the context of the customer problem. Further, we present the results of the classification (stage 1) and the results of the interview analysis (stage 2).

### A. IoT System Description

The IoT solution relies on a distributed system architecture, with relation to different stakeholders: customer, solution provider, cloud infrastructure provider, sensor gateway provider and sensor provider. The mission of the IoT solution can be stated as follows:

*Generate and provide condition monitoring data of a physical customer asset to increase efficiency in managing it.*

## B. Situation

During operations the solution provider was not able to perceive faults in the effect chains and the instantiation of a related failure of the complex IoT system. The customer notification of failure was the only indicator of faults. The customer informed the organization, calling the service desk, that a required function had failed. The incident-specific problem solving procedure was not initiated before customer notification. The problem solving procedure ran by the company, includes the initiation of fault reactions to transfer the system into a controlled state. A team of specialists, representing the stakeholders interest, was executing postmortem analysis, applying the method of 5xWhy to assign root causes. In the analysis the stakeholders applied our classification for anomalies to communications.

## C. Classification Results - Stage 1

In the following, we describe the results of the case study. We applied the classification for anomalies to differentiate between the handling of:

- first, fault detection and fault reaction in the context of situation management from
- second, defect and its options for defect correction in the context of defect management.

### Problem:

- Customer view: customer does not receive up-to-date asset condition information. Loss of control, with risk of damage of customer asset.
- Solution provider view: customer perceives that solution provider lacks competency to control the situation. The relationship of trust between customer and solution provider is at risk.

### Failure:

- Customer (**functional system view**): failure of solution. Information related to asset condition inconsistent due to message transmission fault.
- Provider (**functional system view**): failure of load balancer (single point of failure) with fail passive due to load test on cloud infrastructure. The load balancer failed by a temporary overload and reacted as expected by switching to its controlled state: The system is down. The server and its service is not available.

### Fault (→ fault reaction):

- Customer (**dynamic system view**) : Customer is only offered functional system view. Complexity of dynamic system view is hidden from the customer.
- Provider (**dynamic system view**): failure and fault detection by customers via emergency call to service desk. The organization did not detect intermittent transaction message faults. In consequence, there was no fault reaction. Both, technical (redundancy) as well as an organizational (manual) fault reaction were missing. Incident-based problem solving process did not start until customer calls or sends an email to the service desk.

### Defect (→ option(s) for defect correction):

Defects related to IoT system (**system structure view**):

- Defect 1: malconfigured load balancer.
- Defect correction option 1: correction of load balancer malconfiguration.

Defects related to situation management capability (**system structure view**):

- Defect 2: missing technical option for fault reaction of load balancer (single point of failure): no failing active and operational, by activating a redundant component.
- Defect correction option 2: implement failing active and activating redundancy.
- Defect 3: missing organizational options for fault reaction: no tactical options for situation management. On organizational level solution provider did not agree controlled states for degrading the value proposition.
- Defect correction option 3: implement degrading value proposition including customer information of quality degradation.

## D. Discussion - Stage 1

With the application of the classification, we identified:

- by differentiating the concepts fault and failure, we improve in engineering and operating fault tolerant systems.
- by differentiating the concepts fault and defect, after effective fault fixing, we are able to assign a defect to a component and identify its options for defect correction for transfer to defect management. This is done in efficiency mode to run for efficiency yield. We are customer oriented. Customer interest is attached to fault fixing and not to defect correction and decreasing technical dept. We identified that motivation for defect correction has to be provided by the organization.
- by differentiating the dynamic system view, we are able to assign defects related to IoT system and defects related to capability of situation management.
- during postmortem analysis, we tended to assign the defect to a single component and not to an interacting subset of components. Therefore, we forgo options for correction, with higher potential of efficiency yield and sustainability.
- a need to increase observability in the dynamic system view with a focus on business process to control adding value to the customer.
- for having appropriate tactical options in situation management, additional defects (related to situation management) can be assigned to a subset of components.

In this case the correction of load balancer malconfiguration (defect 1) covers defect correction and fault fix. Defect 2, to eliminate the single point of failure is transferred to backlog. Improvement to increase fault tolerance as well as improvement to increase capability for situation management have been identified and are transferred to defect management. The defect management decides whether and which options for correction are to be executed.

## E. Interview Results - Stage 2

This section provides the results of the participant interviewees. Following the structure of the interview guide, we present the highest level of the coding system with exemplary statements from the stakeholders (S1, S2 and S3) of the postmortem analysis.

**Purpose of classification:** S1 and S2 outlined from experience with other postmortem analyses that discussions between different organizational units are often very imprecise if no shared concepts exist. A classification for anomalies allows to competently exchange about anomalies with teams from other organizational units *"[...] and ensures that you have a common view on the situation and the system."* (S2). Further, a commonly accepted classification, acts as *"[...] a common language [...]"*, that enables to *"[...]improve mutual understanding."* (S2). It is especially important in order to enable continuous learning for iterative product development and improvement along different organizations. *"Learning for 'better' is done by excluding what you do not want to have. Learning for 'better' is done by excluding the anomalies. The challenging communication about anomalies across organizations is made easier by a common classification scheme."* (S3).

**Effectiveness:** All three stakeholders confirmed that it is possible to differentiate between the handling of fault detection and fault reaction from the handling of defect and its options for defect correction. S2 stated that the classification enables to achieve a high accuracy in differentiation. It *"[...] enables to distinguish more precisely between 'quick fix' and sustainable correction."* (S1). S1-S3 conclude, that the current organizational framework of the postmortem analysis is not yet an environment that is sufficiently conducive to learning but the substantive goal was achieved. *"The questioning of 5xWhy, was just limited compatible to it. [...] It has disrupted the regular procedure of the postmortem analysis. Which is why it was not as efficient and took longer. But in terms of content it was definitely valuable."* (S1).

**Efficiency:** The interviewees described that the application of the classification across different stakeholders and organizations as challenging and complicated. *"You have individual views and a global view, which is presented simplified in retrospective. But it is not that easy in the analysis process... in reality it is nested with different responsibilities"* (S2). As the efficiency was limited due to organizational frame, we further discuss the relationship between the efficiency of applying the classification to the procedure of asking 5xWhy, mentioned by all three stakeholders. S3 reflected that the application of the 5xWhy method goes beyond the boundaries between the effect chain and the tactical fault reactions during operation, without explicitly differentiating them. As a consequence: *"With troubleshooting, outstanding unwanted effects 'faults' are reacted to with a fault reaction. When these unwanted effects are under control through a fault reaction, an interest in further actions related to sustainable correction decreases, even if the fault reaction has to be operated persistently."* (S3).

In terms of time consumption our classification performs

worse than the previous postmortem. However, the application of our classification enabled to identify faults with fault reactions and defects with options for sustainable correction. Therefore, defects related to product quality and defects related to capability of situation management could be identified. From a strategic business point of view, the concepts of anomalies consequently allow to weigh up in present and also at a later stage in which activities are to be invested.

**Satisfaction:** All of the three stakeholders of the postmortem analysis stated to have a positive attitude towards the logic of the classification. S1 describes to feel secure in applying our concepts of anomalies and moreover to further distribute it across his team. S3 commented the question of satisfaction by relating it to cultural aspects of performing postmortem analysis: *"What I particularly like, is the fact that the classification refers to anomalies in the IoT system and not to people who are to blame for something."* S1 and S3 reflected on feeling confident in the application of the concepts.

**Further Optimization:** We identified that the classification can be a foundation for an analysis that is not focused on blaming. However, it is not the solution to cultural problem (S1). S2 also discussed this point and stated: *"If you disassemble everything in detail and you take a close look at what went wrong and where a defect is located, that is what people do not really enjoy. At this point you have to be especially careful not to be destructive."* Furthermore, S3 reflected how the execution of the 5xWhy in combination with the classification could be improved in terms of efficiency (S3): *"Once the defect is detected, so as assigned to a subset of components in arrangement, asking 'why' can be stopped. Which option for correction is finally executed, can be strategically planned outside of the postmortem analysis meetings, in the efficiency paradigm. Currently, in the postmortem meetings the options for correction are 'quickly found' regardless of whether these corrections find a higher strategic use."* It turned out that further improvement on training the classification, including practical instructions on how to apply it on a real-world customer problem are necessary.

## F. Discussion - Stage 2

The answers of the participants indicate, if the adaptation of the classification for anomalies is established and sufficiently practiced, it has the potential to be not only effective but also efficient. Misunderstandings in communications across different organisational units or silos can be drastically reduced. In addition, the reusability of the results of the classification fosters a continuous learning culture for iterative and inductive product development and operation. We conclude that there is a need to focus on the organizational frame and on training of our classification for anomalies.

## VI. THREATS TO VALIDITY

Since we performed interviews for evaluating the quality in use of our classification, we anticipate some personal bias in the answers from the interviewees as threat to **internal validity**. To minimize this threat, we triangulated the answers



of one interview of each stakeholder group. Further, the results of the classification have been classified and confirmed by the different stakeholders. Besides, we compared the interview data with the observation of the classification meetings.

Concerning **external validity**, we identified a risk that the classification and the results are specific to the case company context of Bosch. Hence, as we documented our adaptation on the IEEE Std. 1044-2009 for the system level as well as described the case context, we assume that our results can be generalized to similar IoT contexts outside of this case.

## VII. CONCLUSION AND FUTURE WORK

Building and operating IoT systems in open system context needs iterative and inductive learning to improve products. This is enabled by excluding unwanted effects in dynamics of customer context of use by fault reaction or defect correction. To collaborate along different product teams and organizations an ontology providing concepts of anomalies is needed. With this work, we proposed and applied an adaptation of the IEEE Std. 1044-2009 for classifying system anomalies according to three architectural views: functional, dynamic and system structure view. Our adaptation allows to differentiate between the handling of fault detection and fault reaction in the context of situation management from the handling of defects and their options for defect correction in the context of defect management. We have evaluated the quality in use with the focus on effectiveness, efficiency and satisfaction of our adaptation by applying it to a postmortem analysis at Robert Bosch GmbH.

In terms of **effectiveness** we were able to differentiate between the concepts of failure and fault. This enabled us to identify missed fault reactions for controlling the system to keep the customer's trust during situation management and lets us improve fault tolerance and capability for situation management in future. Moreover, we were able to differentiate between the concepts of fault and defect. This enabled us a differentiation between the actions for tactical fault fix from actions for sustainable defect correction. In addition, the results of the postmortem analysis are available in a reusable scheme and allow us to decide whether and how defects have to be corrected. The **efficiency** of applying our classification was limited. The stakeholders described the application of the classification across different organizations as challenging, also due to the organizational frame. However, all interviewees highlighted the valuable content-related contribution. We anticipate that the effort and time of applying the classification will decrease with increasing usage to further improve efficiency. The overall **satisfaction** of applying the classification was positive. The stakeholders pointed out the constructive usage of the concepts, in order to foster continuous learning. The classification can be foundation but will not be solution to difficulties in terms of silo mentality.

We identified that in industry there is a focus on the functional system view on the "happy path". A dynamic system view, to control temporal effects, which enables fault detection and fault reaction is often missing. In order to

sufficiently react to anomalies and keep the system in a controlled state performing operations, we have to observe and quantify the dynamic behaviour to indicate and detect faults. Therefore, we work on methods for observing anomalies to indicate and detect faults along the component-effect chains of a service throughout different organizational units, performing development and operations. This should allow us to improve fault tolerance of complex IoT systems.

## REFERENCES

- [1] S. Niedermaier, F. Koetter, A. Freymann, and S. Wagner, "On observability and monitoring of distributed systems—an industry interview study," in *International Conference on Service-Oriented Computing*. Springer, 2019, pp. 36–52.
- [2] T. Ohno, *Toyota production system: beyond large-scale production*. crc Press, 1988.
- [3] D. Kahneman, *Thinking, fast and slow*. Penguin Books, 2011.
- [4] R. I. Cook, "How complex systems fail," *Cognitive Technologies Laboratory, University of Chicago, Chicago IL*, 1998.
- [5] P. M. Senge, *The fifth discipline: The art and practice of the learning organization*. Broadway Business, 2006.
- [6] J. W. Forrester, "System dynamics, systems thinking, and soft or," *System dynamics review*, vol. 10, no. 2-3, pp. 245–256, 1994.
- [7] DIN, "Quality management systems - fundamentals and vocabulary (ISO 9000:2015)," 2015.
- [8] IEEE, "IEEE standard classification for software anomalies (IEEE Std. 1044-2009)," 2009.
- [9] ISO/IEC, "Systems and software engineering – systems and software quality requirements and evaluation (SQuARE) – measurement of quality in use (ISO/IEC 25022)," 2011.
- [10] S. Wagner, "Defect classification and defect types revisited," in *Proceedings of the 2008 Workshop on Defects in Large Software Systems*, ser. DEFECTS '08. New York, USA: ACM, 2008, pp. 39–40.
- [11] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE transactions on dependable and secure computing*, vol. 1, no. 1, pp. 11–33, 2004.
- [12] B. Beyer, C. Jones, J. Petoff, and N. R. Murphy, *Site Reliability Engineering: How Google Runs Production Systems*. " O'Reilly Media, Inc.", 2016.
- [13] ISO/IEC/IEEE, "Systems and software engineering – life cycle management part1: Guidelines for life cycle management (ISO/IEC/IEEE 24748-1)," 2018.
- [14] ISO/IEC, "Systems and software engineering - vocabulary (ISO/IEC 24765)," 2009.
- [15] P. M. Institute, "A guide to the project management body of knowledge (pmbok® guide)," 2008.
- [16] ISO/IEC, "Systems and software engineering – recommended practice for architectural description of software-intensive systems (ISO/IEC 42010)," 2007.
- [17] ISO, "Road vehicles — functional safety (ISO 26262-1:2018)," 2018.
- [18] IEEE, "Standard for software engineering – software life cycle processes – maintenance (IEEE Std 14764-2006)," 2006.
- [19] J. R. Boyd, "The essence of winning and losing," *Unpublished lecture notes*, vol. 12, no. 23, pp. 123–125, 1996.
- [20] S. Dekker, E. Hollnagel, D. Woods, and R. Cook, "Resilience engineering: New directions for measuring and maintaining safety in complex systems," *Lund University School of Aviation*, 2008.
- [21] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical software engineering*, vol. 14, no. 2, p. 131, 2009.
- [22] ISO, "Ergonomic requirements for office work with visual display terminals (vdt) – part 11: Guidance on usability (ISO 9241-11)," 1998.
- [23] S. Niedermaier, S. Heisse, and S. Wagner, "Interview guide: Quality in use of anomaly classification," 2019. [Online]. Available: <http://doi.org/10.5281/zenodo.3636111>
- [24] P. Mayring, "Qualitative content analysis: Theoretical foundation, basic procedures and software solution," 2014.