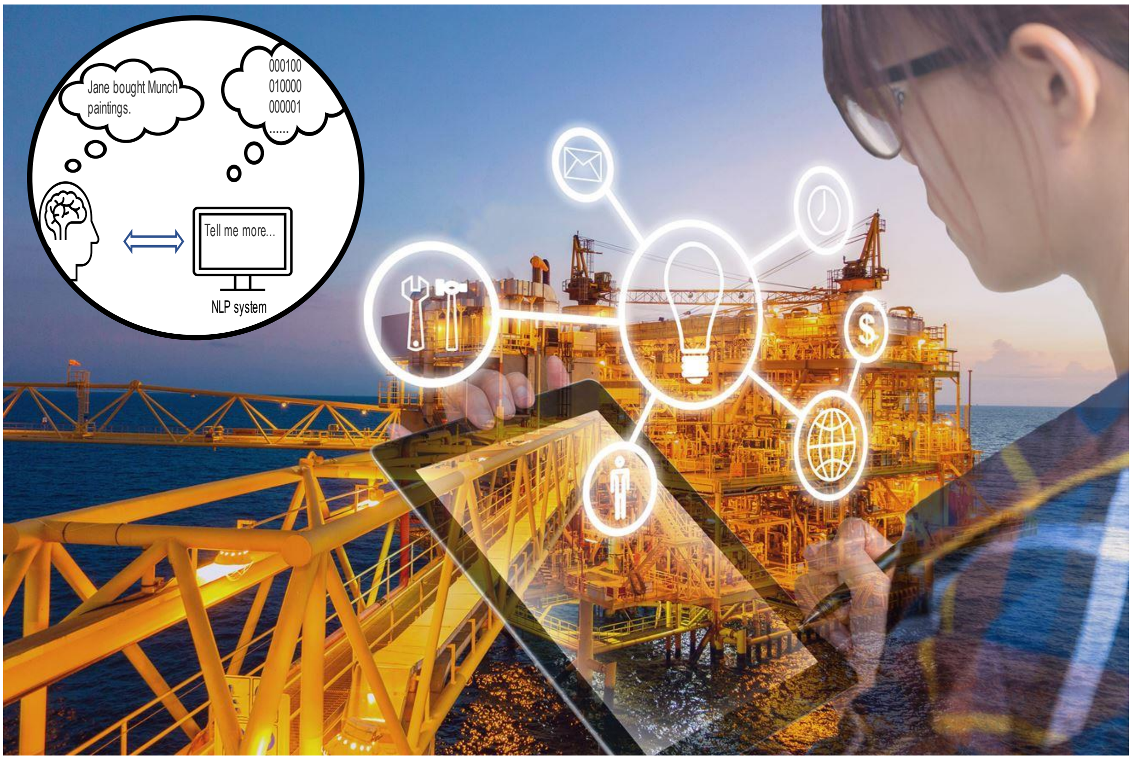


Report

Potential for automated follow-up of safety equipment

An APOS project report

Author(s):

Shenae Lee, Maria Vatshaug Ottermo, Stein Hauge, Solfrid Håbrekke, (SINTEF), Mary Ann Lundteigen (NTNU)

Report No:

2023:00110

Client(s)

Multiclient

SINTEF Digital
Postadresse:
Postboks 4760 Torgarden
7465 Trondheim

Sentralbord: 40005100

info@sintef.no

Foretaksregister:
NO 919 303 808 MVA

Report

Potential for automated follow-up of safety equipment

KEYWORDS
Failure detection
Failure reporting
Failure class determination
Technical Language Processing
Follow-up of safety systems

VERSION	**DATE**
01	2023-03-10

AUTHOR(S)
Shenae Lee, Maria Vatshaug Ottermo, Stein Hauge, Solfrid Håbrekke, (SINTEF), Mary Ann Lundteigen (NTNU)

CLIENT(S)	**CLIENT'S REFERENCE**
Multiclient	Erik Korssjøen
PROJECT NO	**NO. OF PAGES/APPENDICES**
102020273	46

SUMMARY
On petroleum facilities, safety equipment is installed to obtain risk reduction. To ensure that the desired risk reduction is achieved during the operational phase, follow-up of the reliability of such equipment is important. Essential means to follow up safety-critical equipment are the reporting, classification and analysis of failure data. However, experience shows that considerable manual effort is needed to attain high-quality data, and therefore, this report investigates possibilities for automated failure reporting and failure class determination. The proposed approaches involve the potential to guide the user with pre-defined help-texts for improved failure reporting, the possibility to utilize relevant information from systems such as the safety and automation system etc. for automatic determination of failure mode and detection method, the potential to automate failure class determination based on standardized failure mode and detection method hierarchies, as well as the use of technical language processing (TLP) methods to extract key information for improved and efficient failure class determination.

PREPARED BY Shenae Lee, Maria Vatshaug Ottermo, Stein Hauge	SIGNATURE SHENAE LEE SHENAE LEE (Mar 14, 2023 10:49 GMT+1)
CHECKED BY Lars Bodsberg	SIGNATURE Lars Bodsberg
APPROVED BY Anita Øren	SIGNATURE Anita Øren

REPORT NO.	**ISBN**	**CLASSIFCATION**	**CLASSIFICATION THIS PAGE**
2023:00110	978-82-14-07942-5	Unrestricted	Open

COMPANY WITH
MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO 9001 • ISO 14001
ISO 45001

Document history

VERSION	DATE	VERSION DESCRIPTION
0.1	2022-11-11	Draft
01	2023-03-10	Final

Table of content

Preface

The current work has been carried out as part of the APOS research project "Automated process for follow-up of safety instrumented systems (SIS)" (*Norw: Automatisert prosess for oppfølging av instrumenterte sikkerhetssystemer*). The project is supported by the Research Council of Norway and the APOS and PDS[1] members. The project duration has been 2019-2022. We would like to thank everyone who has given input and comments to the report, and who has participated in numerous meetings, seminars, and workshops.

The main purpose of the APOS research project has been to simplify and standardize reporting and classification of SIS failures, including the classification of safety equipment, and to provide a basis for increased automation and standardisation of SIS follow-up, including a specification for an information model for functional safety. The APOS project comprises seven related activities:

1. H1: Guidelines for standardised equipment classification and failure reporting [1]
2. **H2: Potential for automated follow-up of safety equipment (this report)**
3. H3: Guideline for follow-up of Safety Instrumented Systems (SIS) in the operating phase [2]
4. H4: Standardised/electronic SRS format [3]
5. H5: Information model for functional safety [4]
6. H6: Project summary and presentation
7. H7: PDS Data handbook, 2021 Edition [5]

This report documents project activity two (H2).

Trondheim, March 2023

[1] PDS is a Norwegian acronym for reliability of safety instrumented systems. For more information about PDS and the APOS project, reference is made to www.pds-forum.com

1 Introduction

1.1 Background and objective

This guideline documents activity two (H2) in the APOS research project "Automated process for follow-up of safety instrumented systems (SIS)" (*Norw: Automatisert prosess for oppfølging av instrumenterte sikkerhetssystemer*). A main purpose of this project is to simplify and standardise reporting and classification of SIS failures, including the classification of safety equipment, and to provide a basis for increased automatisation and standardisation of SIS follow-up.

As per today, follow-up of safety systems requires considerable manual resources. Reporting and classification of maintenance and failure data is often subject to concerns about the adequacy, quality, and uncertainty of the data. An important starting point for addressing these concerns is therefore to ensure that failures are reported in a consistent way, with a high level of precision about failure mode and detection method. A main objective of the H2 activity is to contribute towards further digitalization, focusing on automated reporting of failure mode and detection method and determination of failure class.

1.2 Abbreviations and terminology

Below, abbreviations applied in this report are explained.

AIR	Anormal instrument reading
ASR	Automatic shutdown report
APOS	Norwegian acronym for Automated process for follow-up of safety instrumented systems
CCF	Common cause failure
CCR	Central control room
CMMS	Computerized maintenance management system
DD	Dangerous detected
DEX	Defect of EX protection
DOP	Delayed operation
DSE	Degraded sensing
DU	Dangerous undetected
ELP	External leakage process medium
ELU	External leakage utility medium
ERO	Erroneous output
ESD	Emergency shutdown
FTC	Fail to close
FTO	Fail to open
FTS	Fail to start
F&G	Fire and gas
HIO	High output
IEC	International Electrotechnical Commission
IMS	Information management system
ISO	International Organization for Standardization

LEL	Lower explosion limit
LCP	Leakage in closed position
LEX	Loss of EX protection
LOC	Loss of containment
LOO	Low output
NER	Named entity recognition
NLP	Natural language processing
NOI	Noise
NONC	Non-critical
NOO	No output
OT	Operational technology
PFD	Probability of failure on demand
PM	Periodic maintenance
PSA	Petroleum Safety Authority (Norway)
PSD	Process shutdown
RNNP	Trends in risk level in the petroleum activity in Norway
S	Safe
SAP	Systems, applications, and products (Software)
SAS	Safety and automation system
SER	Minor in-service problems
SF	Safe failures
SFI	Safety function impaired
SIL	Safety integrity level
SIF	Safety instrumented function
SIS	Safety instrumented system
SPO	Spurious operation
SRS	Safety requirement specification
QA	Quality assurance
TLP	Technical language processing
VIB	Vibrations

Table 1: Terminology (to be updated in final version)

Common cause failures (CCF)	Failures of multiple items, which would otherwise be considered independent of one another, resulting from a single cause (ISO 14224:2016) [6].
Corrective maintenance	Maintenance carried out after fault detection to effect restoration (ISO 14224:2016) [6].
Dangerous failure	A failure that impedes or disables a given safety action (IEC 61511:2016) [7]. *APOS comment: A fraction of these failures will be revealed by automatic diagnostic tests and are denoted dangerous detected failures. The residual dangerous failures, not detected by self-tests, are denoted dangerous undetected failures.*

Dangerous detected failure	A dangerous failure revealed by automatic diagnostic tests and provided with alarms or in such a way that the failure is detected "immediately". E.g., a blocked beam alarm from an optical line gas detector.
Dangerous undetected failure	Dangerous failures not detected automatically upon occurrence but revealed only by a functional test or upon a demand (PDS data handbook) [5]. NOTE: A demand can both be a planned shutdown / activation / maintenance activity or a casual/unplanned activation/observation of the equipment.
Degraded failure	Failures where the ability of the equipment to carry out the required safety function (or maintain production) has not ceased but is reduced, and which over time may develop into a critical failure (PDS data handbook, 2021) [5].
Detection method	Method or activity by which a failure is discovered (ISO 14224:2016) [6].
Equipment group	Group/class of equipment units with comparable characteristics (function and/or design). All elements within an equipment group normally have a comparable failure rate.
Failure	<of an item> loss of ability to perform as required (ISO 14224:2016) [6].
Failure cause (root cause)	Set of circumstances that leads to failure. Note 1 to entry: A failure cause can originate during specification, design, manufacture, installation, operation, or maintenance of an item (ISO 14224:2016) [6].
Failure mechanism	Process that leads to failure. Note 1 to entry: The process can be physical, chemical, logical, or a combination thereof (ISO 14224:2016) [6].
Failure mode	Manner in which a failure occurs (ISO 14224:2016) [6].
Failure rate	Conditional probability per unit of time that the item fails between *t* and *t+dt*, provided that it has been working over [0,*t*] (ISO 14224:2016) [6].
Functional test	Activating an intended function and comparing the response against the requirements. Planned operation performed at constant time interval in order to detect the potential hidden failures which may have occurred in the meantime, e.g., by activating an intended function and comparing the response against the requirements (ISO/TR 12489:2013) [8]. NOTE: Such tests are often named "proof test" in functional safety standards (see below).
Hidden failure	Failure which is not immediately evident to operations and maintenance personnel. Hidden failures do not show themselves when they occur. The hidden failure may be revealed by periodic tests/inspections, casually or when the function of the item is required (on demand) (ISO/TR 12489:2013) [8].
Natural language processing	An area of research in computer science and artificial intelligence concerned with processing natural language like English. [9]
Named entity recognition	Named entity recognition is a technique for information extraction in natural language processing. A named entity is a word or a chunk of words that contains a certain type of

	information, for example Olav, Trondheim, and Tesla. A named entity can be annotated or tagged with an entity type such as person, location, and company [10].
Non-critical failure	Failure of an equipment unit that does not cause an immediate cessation of the ability to perform its required function. (ISO 14224:2016) [6]. The main functions of the component are not affected. Examples may be sensor imperfection or a minor leakage of hydraulic oil from an actuator, which has no immediate impact on the specified safety function. These failures correspond to the no-effect failures. (IEC 61508:2010) [11].
Preventive maintenance	Maintenance carried out at predetermined intervals or according to prescribed criteria and intended to reduce the probability of failure or the degradation of the functioning of an item (ISO 14224) [6].
Proof test	Periodic test performed to detect dangerous hidden failures in a SIS so that, if necessary, a repair can restore the system to an 'as new' condition or as close as practical to this condition (IEC 61511:2016) [7]. *APOS comment: If the proof test can detect all dangerous hidden failures, the proof test coverage is 100%. If the proof test is not able to detect all dangerous hidden failures, the proof test coverage is less than 100%.*
Random hardware failure	Failure, occurring at a random time, which results from one or more of the possible degradation mechanisms in the hardware (IEC 61511:2016) [7].
Reliability	Ability of an item to perform a required function under given conditions for a given time interval (ISO 14224:2016) [6].
Reliability data	Data for reliability, maintainability and maintenance support performance. (ISO/TR 12489).
Safe failure	Failure which favours a given safety action Note 1 to entry: A failure is "safe" only with regard to a given safety function (IEC 61511:2016) [7].
Spurious failure	Unexpected shutdown resulting from failure(s) in the control/monitoring system or error(s) imposed on the control/monitoring system originating from the environment or people (ISO 14224:2016) [6].
Systematic failure	Failure related to a pre-existing fault, which consistently occurs under particular conditions, and which can only be eliminated by removing the fault by a modification of the design, manufacturing process, operating procedures, documentation or other relevant factors (IEC 61511:2016) [7].
Tag number	Unique code for each equipment unit, indicating the type of equipment and the system it belongs to.
Technical language processing	Term Technical Language Processing is proposed to refer to NLP applications for engineering texts in a scalable and reproducible way [12].

1.3 Structure of the report

Chapter 2 introduces some basic information about failure reporting and failure class determination, including why it this important, how it is performed, and which challenges are common today. Based on the challenges, chapter 3 considers possible ways to aid the user while performing failure reporting and failure class determination, by providing pre-defined help-texts or automated determination of failure mode, failure method and failure class based on information from other systems. Furthermore, chapter 4 looks into the possibility to categorize text-based data in a consistent way using technical language processing (TLP). Chapter 5 combines the approaches of chapters 3 and 4 to further explore the potential of automated failure reporting and failure class determination.

2 Failure reporting and failure class determination

2.1 Importance of failure reporting and failure class determination

In design of oil and gas installations, safety barriers are installed to prevent accidents such as fire and explosion. The performance of these barriers shall be followed-up during operations to verify that the intended barrier functions can be achieved. Follow-up of barriers is required by the Petroleum Safety Authority (PSA) Norway's Management regulations §5 on barriers stating that " Personnel shall be aware of what barriers have been established and which function they are intended to fulfil, as well as what performance requirements have been defined. Personnel shall be aware of which barriers and barrier elements are not functioning or have been impaired". The guideline to this section states that "for safety systems, standard such as IEC 61508, IEC 61511, IEC 62061 and ISO 13849 should be used as a basis. In addition, Norwegian Oil and Gas' Guideline 070 should be used as a basis for offshore petroleum activities".

A SIS is used to perform one or more safety instrumented functions (SIFs), and each SIF will have a performance requirement that is expressed as a safety integrity level (SIL). The required SIL for a SIF is determined based on risk assessment, alternatively by using the minimum SIL requirements from the 070-guideline. For a low-demanded SIF, SIL is assigned based on the probability of failure on demand (PFD) [11]. The PFD is initially calculated in the design phase to verify that the target SIL is met. The PFD of a low-demanded SIF primarily relates to dangerous undetected (DU) failures, and hence the DU failure rate is a main input parameter to this calculation [13]. The failure rate assumed in the design phase, is also called the design failure rate. However, design failure rates may not coincide with the reliability performance during the operational phase. For this reason, failure rates are updated to reflect equipment performance during operations, and to verify the actual risk reduction achieved during operation. This aspect is emphasized in the second edition of IEC 61511-1 (Clause 11.9.3) which states that the applied reliability data (such as the failure rate) shall be *credible, traceable, documented and justified* and that it shall be based on field feedback from similar devices used in a similar operating environment [7]. To meet these requirements, high-quality reporting and analysis of failure data are essential activities for SIS follow-up.

2.2 How to perform failure reporting and failure class determination

An important starting point for addressing the concerns about quality of maintenance and failure data is to ensure that failures are reported and classified in a consistent way. Figure 1 illustrates a typical workflow for failure event reporting, failure class determination, and further data analysis, with focus on 'Failure reporting' and 'Failure class determination' (indicated with the dotted, red circle).

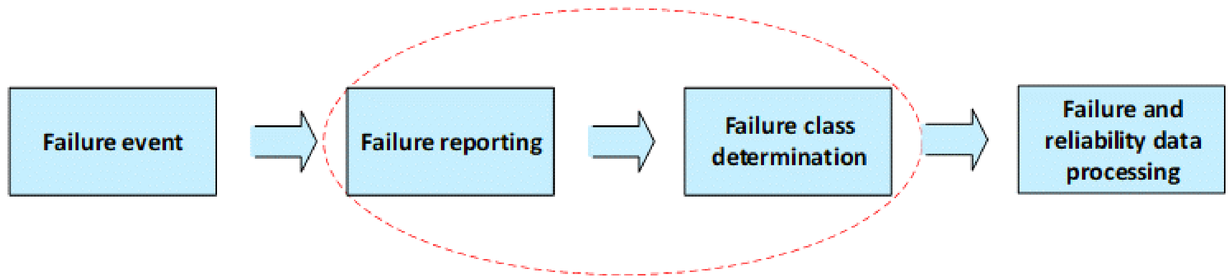


Figure 1: Workflow for failure event and reporting, failure class determination, and further data analysis

Failure data is registered in the Computerized Maintenance Management System (CMMS) by use of notifications and/or work orders. Notifications for safety equipment can be created from scheduled maintenance, unscheduled maintenance, random observations, and real demands. It should be noted that a notification may not necessarily include failure reporting, because it can, for example be created to report findings from preventive maintenance activity from which no failure is discovered [14] or simply as an improvement suggestion. A notification usually contains fields for detection method and failure mode, which is the minimum information required to determine the failure class [1]. A notification will typically consist of a set of data fields, including equipment tag, location, description, detection method, failure mode, etc., and will have a unique notification ID. The data fields can be fixed data fields, drop down fields and free text fields. An example of relevant *parts of* a notification is given in Table 2.

Table 2: Example of notification content

Date	No	Tag no.	Description	Location	Failure mode	Problem code text	Symptom	Long text	Detection method
2021-09-27	123 456	xx.yyy. zz - ##-####	EV not functioning	Slug catcher gas outlet	OTH	Fail to open / close	Design related	* Valve does not close, stops at approx. 15% closed and slowly creeps up to approx. 40% *Failure on ESD and PSD	PM

The **detection method** characterizes how the failure was discovered. If the failure was revealed immediately (i.e. alarmed upon occurrence), it is classified as detected. On the other hand, if the failure was revealed during a functional test, periodic maintenance, causal observation, or demand (i.e. latent until test or demand), it is classified as undetected. In some cases, the detection method field is a free text field instead of drop-down list, where the technician writes a short text about the detection method (e.g. partial stroke testing, PM (Periodic Maintenance etc.)).

The **failure mode** characterizes the manner in which a failure occurs. For instance, for a valve the safety function will typically be to open or close on demand, and to keep tight in the closed position. Hence, a failure to close or open on demand or an internal leakage in closed position are both examples of failure modes. By combining the failure mode with information about the equipment group, and its related safety function, it can be determined whether the failure is dangerous or safe, or non-critical. Hence, correct reporting of failure modes is essential for determining the severity of the failure.

As part of the failure class determination, each notification is classified as either DU or not, see Figure 2.

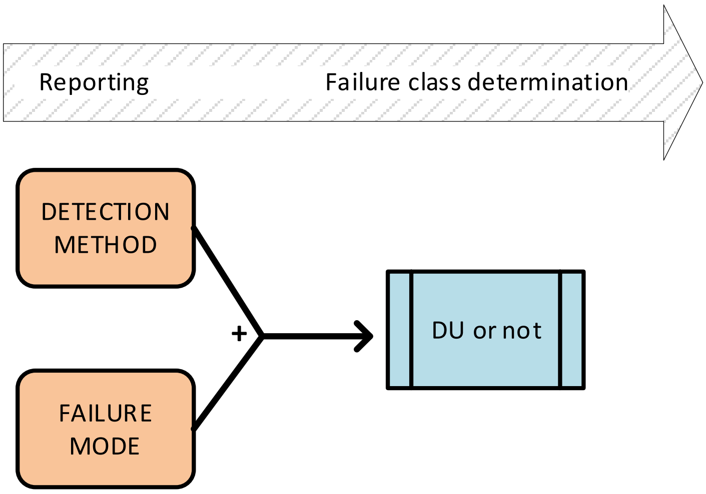


Figure 2: Determination of failure class based on reported detection method and failure mode

2.3 Challenges with failure reporting and failure class determination

As mentioned in Section 2.1, high quality reporting is crucial to enable classification of DU failures. However, the collection of failure notifications and identification of DU failures are time consuming. In addition, the notification quality varies, meaning that essential information to classify failures may be missing, which means that it is often necessary to investigate other sources such as event logs or to involve equipment experts and/or maintenance personnel to be able to determine the failure class. Based on the experience from manually classifying almost 30.000 notifications to obtain the failure rates presented in the PDS data handbook [5], we estimate that classifying 500 notifications takes three to four days, and involves between four and six persons. This amounts to a total of approximately four man-labour years for classifying all 30.000 notifications.

Today, the detection method and failure mode are manually registered in the notifications. In the maintenance systems the parameters can usually be selected from a list of alternatives mainly based on ISO 14224 [6]. ISO 14224 provides a basis for collection of reliability data for oil and gas equipment including equipment groups and lists of possible detection methods, failure modes, failure mechanisms, and failure causes for the equipment groups. The technicians reporting failure notifications are not always familiar with the listed alternatives and the meaning of all possible categories, particularly for the failure mode. As a result, the 'other' or 'unknown' categories are often selected for the failure mode. An internal study performed over a six-month period for a Norwegian offshore facility showed that for more than 50% of the notifications, the failure mode was classified as either 'other' or 'unknown'. It should however be stressed that this practice varies considerably from installation to installation. However, as a result, the most relevant and trustworthy information about failure mode, as well as detection method and failure cause, is often found in the free text field(s) in the notifications.

Another challenge is that notifications are sometimes written for components that are not part of the equipment's main function. For instance, if a notification is written for the main valve when the failure is actually related to the solenoid that is not defined to be within the boundary conditions of the valve, this is not always possible to identify without reading the free text field. Another common example is notifications written for logic solver units that often do not have a unique tag, implying that the notification is registered against the cabinet where the unit is located or some other related component instead.

It has also been pointed out that determining the correct detection method category can be difficult. As a result, some companies have implemented simplified schemes, e.g., to classify failures as either "hidden" or "revealed". The simplified category suits the purpose of splitting between detected and undetected failures, rather than identifying the actual detection method itself [1].

3 User guided and automatic failure reporting and failure class determination

As already mentioned, failure information in maintenance notifications is usually entered manually by the maintenance technicians. The failure class determination process is also manual. In this chapter we explore the potential for more user aided and/or automatised pre-filling of the notifications as well as automated failure class determination. Three related approaches are discussed:

1. **User guidance on selection of failure mode and detection method:** Additional pre-defined help-text (or algorithms) based on information or logic from other relevant systems (SAS, IMS etc.) or tools *to aid* the maintenance personnel when filling in failure mode and detection method manually. The provision of automatic user guidance can be useful in aiding the correct choice of parameters, e.g., by using text pop-ups, mouse-over etc. Such help texts[2] could e.g., appear in the maintenance system where the failure notification is filled in, secondary in the operational procedures accompanying failure reporting and failure class determination[3].

2. **Automatic determination of failure mode and detection method:** Additional information from relevant OT systems such as SAS, IMS, CMMS that can be used for automatic determination of failure mode and detection method, e.g. for *pre-filling* directly into the maintenance notifications.

3. **Automatic determination of failure class (DU, DD, S)**: e.g. undetected versus detected failure in combination with dangerous versus safe failure.

These approaches are further discussed for the failure mode and detection method. The *combination* of these approaches aided by technical language processing (ref. chapter 4) is discussed in chapter 5.

3.1 Relevant source systems: IMS, SAS, CMMS, etc.

The following section provides a short description of relevant source systems that can give additional information that can be used for automatic determination of failure mode and detection method.

Computerized Maintenance Management System (CMMS)

Operator companies report, classify and document the condition and failure of equipment discovered during operation, testing and maintenance, in a computerized information and management system. A typical example on the Norwegian continental shelf is the SAP maintenance system. Each observation is saved as a notification linked to an equipment tag, which is a unique physical identification tag mounted on the equipment. The CMMS also contains all the maintenance history that can be applied to support classification.

[2] To ensure optimal use and understanding of the help texts, they should not be too long (although not directly comparable, research into getting optimal attention on social media such as Facebook; Twitter, LinkedIn, Instagram etc., indicate not more 100-150 characters, i.e., in the order of 18-25 words: https://www.verygoodcopy.com/hootsuite-1/the-ideal-social-media-post-length-a-guide-for-every-platform)

[3] More detailed explanations and examples concerning the understanding of e.g., the taxonomy, should also be a part of the training material.

Condition monitoring systems
Systems for condition monitoring continuously collect data about condition of equipment such as vibration and temperature. These systems typically provide alarms in case of a degraded state of the equipment which can provide a basis for decisions about the necessary maintenance of equipment. Valve watch is an example of a condition monitoring system for valves.

Safety and Automation System
System to control and monitor safety instrumented functions, production and processing functions, power distribution functions, ballast and bilge functions and marine system functions [15].

Information Management System (IMS)
An Information Management System (IMS) refers to system that facilitates the integration, processing, organization, and presentation of data from various systems, including SAS, condition monitoring systems etc. The extent and boundaries of the IMS system varies between companies, in some cases the IMS is equivalent to a database, while in other cases it can include both a database and various relevant applications.

Note also that combining information from notifications with information about relevant reliability influencing properties such as for instance dimension, measuring principle, medium, etc. could provide additional insight that can be used both for internal follow-up, feedback to equipment manufacturers as well as improved quality of failure data sets. Which properties that can influence the reliability depends on the equipment type, for more details we refer to [1]).

3.2 Automatic determination of detection method

As mentioned in Section 2.3, it can be difficult to select the correct detection method. ISO 14224 has defined ten categories of detection methods, and as per today, most companies apply the ISO 14224 taxonomy (or variants of this) for classifying detection method. However, some companies have implemented simplified schemes, e.g., to classify failures as either "hidden" or "revealed". This simplified categorisation suits the purpose of splitting between detected and undetected failures, but is not suitable for identifying the actual detection method itself [1]. The report "Guidelines for standardised classification and failure reporting for safety equipment in the petroleum industry" [1], suggests a flexible detection method hierarchy that is compatible with ISO 14224 and provides the means to both identify the detection method and split between detected and undetected failures. Based on company needs and preferences, it is possible to either implement several levels of detection methods, e.g., level D1 and D2 as suggested in this guideline, only level D0 (as some companies have already implemented), or only the ISO 14224 taxonomy. For more details, we refer to the report [1], but for improved readability, the overall structure is rendered in Figure 3 and Table 3 below.

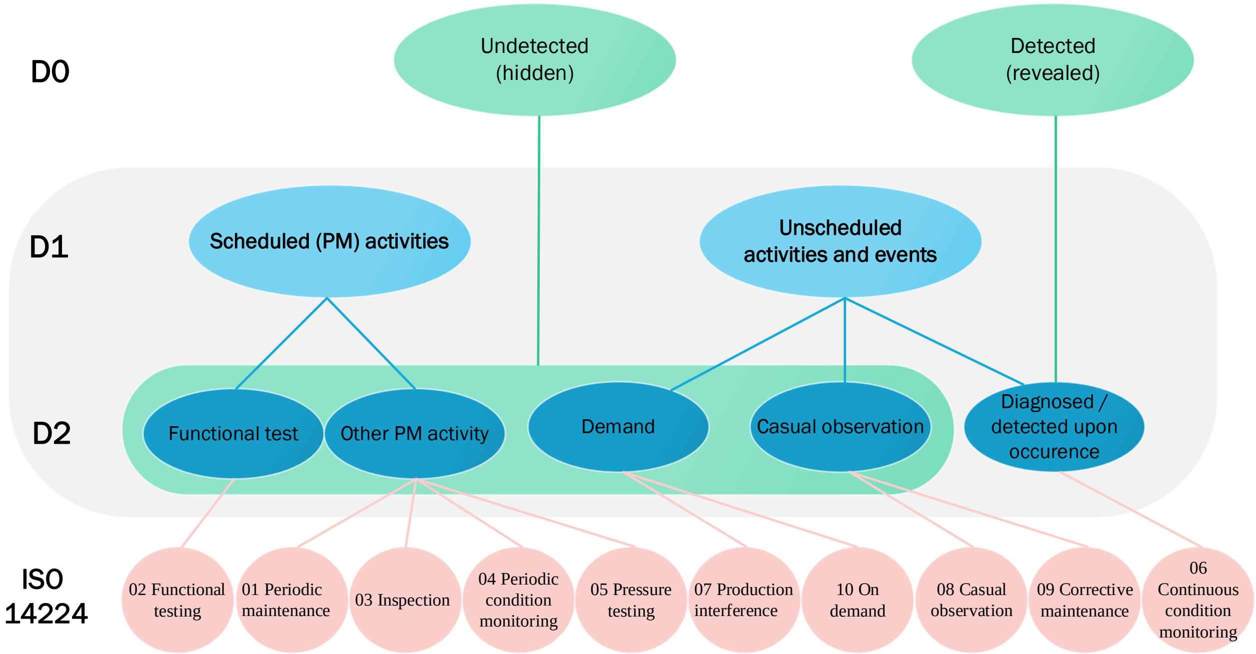


Figure 3: Suggested (greyed-out) detection method hierarchy and mapping towards ISO 14224 alternatives [1]

Table 3: Suggested detection method hierarchy [1]

D0	Detection method (D1)	Detection method (D2)	Examples	Corr. ISO 14224 categories (cf. ISO Table B.4)
Hidden / Undetected	1. Scheduled activities	1.1 Functional test	• Proof test / PM test / SIL test • Leakage test of shutdown valve • Partial stroke test of shutdown valve • Proof test of gas detector	02 Functional testing
		1.2 Other PM activity	• Periodic overhauling / service • Planned periodic inspections and walkarounds • Planned activations between testing (e.g., planned periodic activation of fire dampers during night shift) • Periodic condition monitoring (thermography, off-line vibration measuring, oil analyses, etc.) • Preventive maintenance preparations	01 Periodic maintenance 03 Inspection 04 Periodic condition monitoring
	2. Unscheduled activities and events	2.1 Demand	• ESD / PSD trip (review of automatic shutdown reports, event logs, physical checks in field) • Other demands / activations (e.g., operation of valve, start of pump, start of emergency generator, closure of fire door, resets after shutdown, minor hydrocarbon leak activating nearby gas detector) • During production upsets and instabilities (and associated lack of demand response from	07 Production interference 10 On demand

D0	Detection method (D1)	Detection method (D2)	Examples	Corr. ISO 14224 categories (cf. ISO Table B.4)
			equipment, e.g., no level transmitter response on high level)	
		2.2 Casual observation	• Failure revealed casually when working on / maintaining or preparing maintenance on *other / nearby* equipment • *New/additional* failure on same item revealed when performing repair/CM • Casual observation during production (e.g., attempt to close valve during re-routing of production) • Unplanned walkaround checks • Casual observation on screen (without diagnostic alarm), e.g., manual comparison of process transmitters from central control room (CCR) • Casual observation from monitoring logs / event logs	05 Pressure testing [1)] 08 Casual observation 09 Corrective maintenance
Revealed / detected	**3. Alarmed upon occurrence** [2)]	**3.1 Diagnosed / immediately detected event**	• Self-test/diagnostic alarm (e.g., line gas detectors give alarm if beam is blocked) • Online and immediately alarmed comparison of instruments • Immediate detection by continuous condition monitoring (HART alarms, etc.) • Immediate detection of physical damage (e.g., detector out of location during material handling)	06 Continuous condition monitoring

3.2.1 Approach 1: User guidance on selection of detection method

Help text to level 1 (D1) taxonomy

The following help texts are examples to aid the selection of the D1 detection methods. The short texts should be extendable / possible to elaborate by clicking/ mouse-over:

Scheduled activities

The failure has been revealed during execution of a scheduled or planned activity such as functional testing, maintenance, or inspection…

Examples [*Partial stroke test, Leakage test, Periodic overhauling/service*]

Unscheduled activities and events

The failure has been revealed incidentally, either during or after an unplanned activity or event such as a production shutdown, a production upset, other unplanned demands or activations *or* by casual observations. An example of the latter is that the process operator identifies a leakage from a valve, by observing noise or spill, during other work in the process area.

Examples [*ESD/PSD trip, Demand during production upsets and instabilities, Casual observation*]

Diagnosed/immediately detected event
The failure has been revealed as a result of an automatic alarm in the local or remote SAS/CCR room or other normally manned operating centre where the alarm is given immediate attention. This automatic alarm is caused by self-diagnostic, automatic comparison and/or condition monitoring built into the system.

Examples [*Immediate detection by continuous condition monitoring, Self-test/diagnostic alarm*]

Illustration of help text
Possible use of such help texts and mouse-over functionality is further illustrated in Figure 4. The examples given should be equipment specific (here exemplified by valve failures).

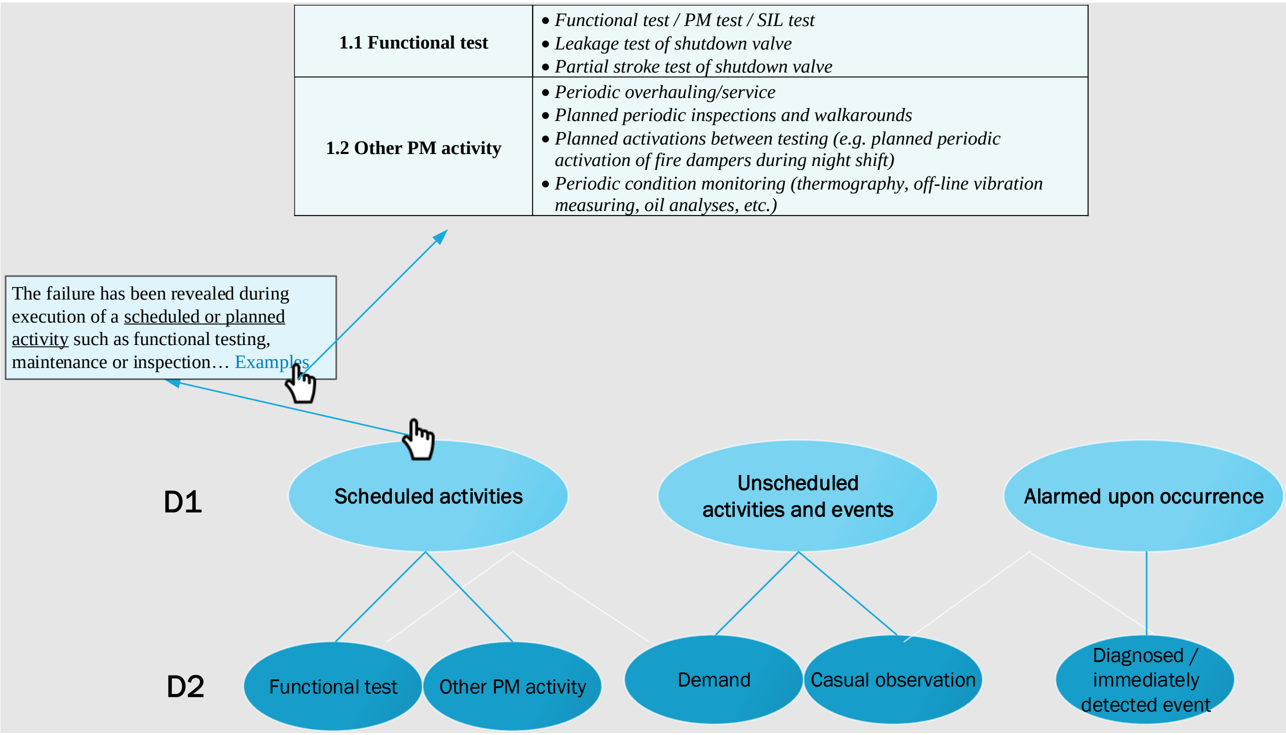


Figure 4: Illustration of possible pop-up text functionality (during valve failure)

3.2.2 Approach 2: Automatic determination of detection method
For failures detected by self-diagnostics, continuous condition monitoring alarms and/or other type of instrument readings, there will be a potential for automatic determination of the detection method. E.g. for line gas detectors that give alarm if the beam is blocked, it can be automatically determined and registered that the detection method was “Self-test, diagnostic alarm”.

3.2.3 Approach 3: Automatic determination of D0-level detection method

The suggested detection method hierarchy also provides a structuring that is suitable for an automated algorithm to determine whether the failure is detected or undetected (D0). The suggested steps are:

1. The user must select one level 1 (D1) category.
2. This selection will then *automatically determine* whether the failure is a "Detected" or "Undetected" failure (D0-level).

3.3 Automatic determination of failure modes

Section 2.3 briefly discusses that choosing the right failure mode can sometimes be challenging for the technicians. The reason is often that there are too many choices, especially if the failure modes are chosen from a generic list of failure modes rather than a list restricted to the possible failure modes of a specific equipment. In [1], a taxonomy with limited number of failure modes for each equipment group is suggested, meaning that the taxonomy is equipment specific. The idea is to use the most relevant failure modes for an equipment group to simplify reporting and thereby improve both the amount and quality of failure mode reporting in notifications. As described in more detail in [1], the suggested failure modes have been arranged into two levels; F1 and F2. An illustration of the suggested taxonomy (greyed-out) is shown in Figure 5 (not equipment specific), while an example failure mode taxonomy for gas detectors is given in Table 4.

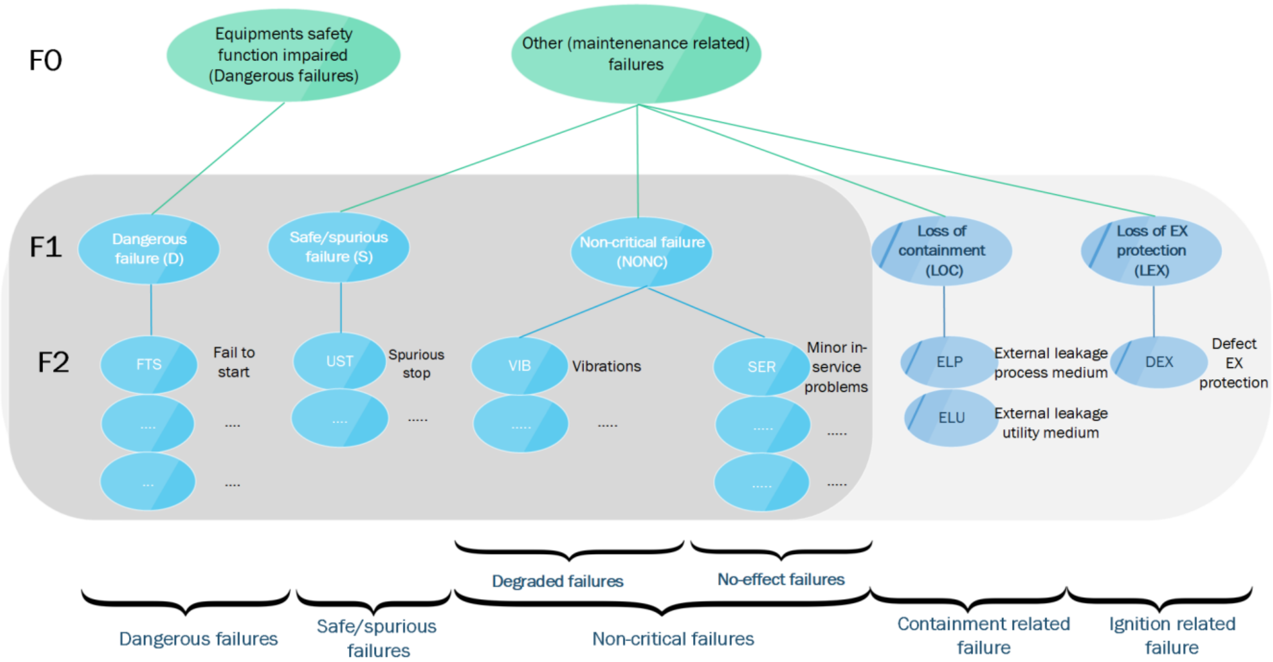


Figure 5: Suggested (greyed-out) failure mode hierarchy – general illustration [1]

Table 4: Failure mode taxonomy for gas detectors

F0	Failure mode (level 1)	Failure mode (level 2)
Dangerous failures	**Safety function impaired**	No output (NOO) [1)]
		Low output (LOO)
		Degraded sensing (dirty optics) (DSE) [2), 3)]
Safe failures	**Safe failure**	Spurious operation (SPO)
Non-critical (NONC) failures	**Degraded failure**	Low output (LOO)
		High output (HIO)
		Erroneous output (ERO)
		Degraded sensing (dirty optics) (DSE) [2), 3)]
	No-effect failure	Minor in-service problems (SER)
	Loss of explosion (EX) protection	Defect of EX protection (DEX)

1) For acoustic gas detectors, only NOO is assumed relevant as dangerous failure mode (SFI).
2) Degraded sensing (dirty optics) is normally a degraded failure mode, but this depends on type of detector and detector configuration.
3) DSE is only relevant for IR/optical gas detectors (not for catalytic gas detectors).

3.3.1 Approach 1: User guidance on selection of failure mode

As seen from Figure 5, the selection of correct failure mode at level 1 (F1) is essential to identify the dangerous failures. To simplify user selection of correct failure mode at this level, a flow diagram / decision tree with example mouse-over help text, has been suggested. This is illustrated in Figure 6.

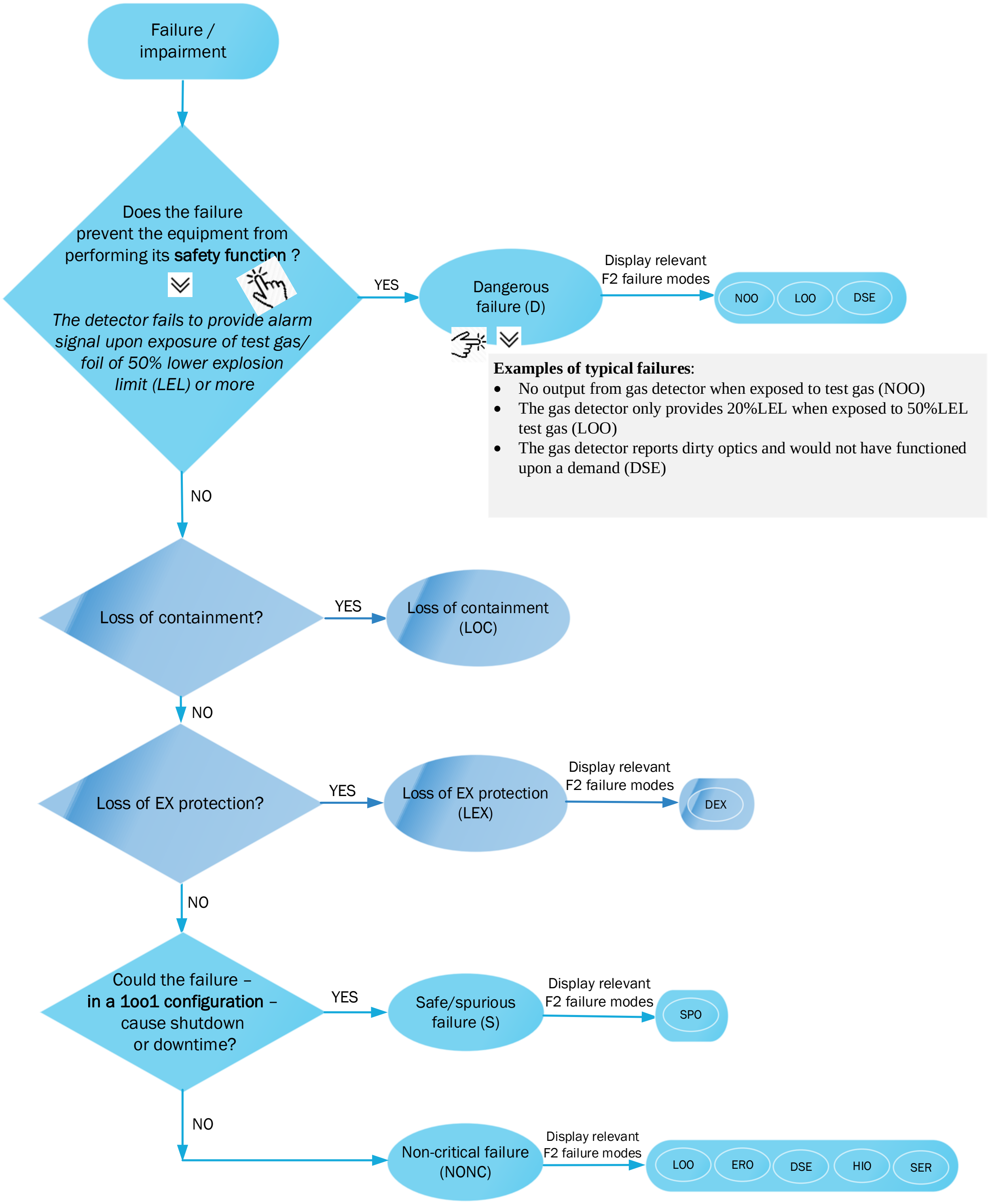


Figure 6: Flow diagram for determination of failure mode at level 1 (IR gas detector example) [1]

Here, the definition of a safety critical failure for the given equipment is obtained e.g., by using mouse-over, illustrated with ☞ in the figure and it is possible to get examples of relevant failures on level 2 (F2) for the specific equipment (here exemplified by typical dangerous failures for an IR gas detector).

A similar flow diagram could be implemented in the maintenance system where the failure modes are registered. For equipment types where e.g., the LOC, the LEX or the SF failure modes are not relevant, the flow diagram can be further simplified.

3.3.2 Approach 2: Automatic determination of failure modes

For failures detected during testing, demands and for failures detected by self-diagnostics, condition monitoring alarms and/or other type of instrument readings, there will be a potential for automatic determination of failure mode. E.g., for shutdown valves, the valve can be set in test mode, and it can automatically be determined whether the valve fails to close (FTC) or whether the response time is excessive (DOP). Similarly, feedback from the limit switches[4] can be used to determine a failure to close or open upon demand and (combined with the cause and effect) to register spurious operations (SPO). Figure 7 illustrates such potential for shutdown valves ("green" failures modes has a potential to be automatically determined and registered in the notification). Note that with additional instrumentation and condition monitoring systems such as acoustic noise detectors (NOI) and valve monitoring of flow characteristics (LCP), additional failure modes *may be* detected (or even predicted in advance).

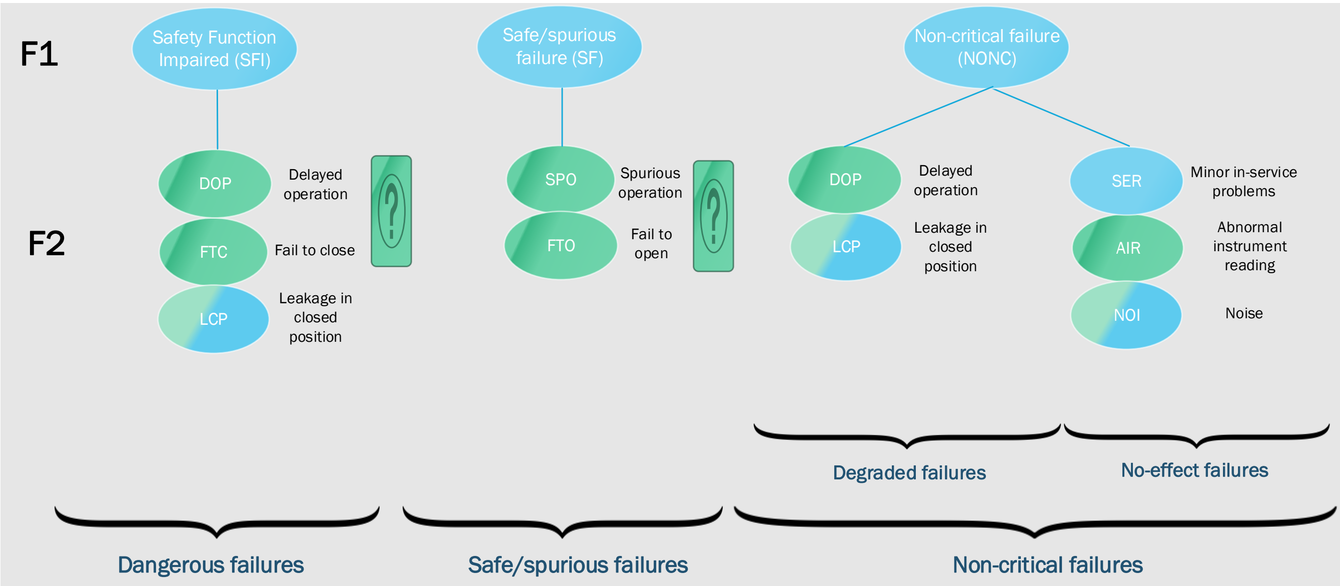


Figure 7: Illustration of potential for automatic determination of shutdown valve failure modes

3.3.3 Approach 3: Automatic determination of F0 level failure modes

As seen from Figure 5 and Table 4, the failure modes for each equipment group can, at level 1 (F1), be categorised into dangerous, safe/spurious and non-critical modes of failure (in addition to LOC and LEX).

[4] Given that the limit switches functions as they shall

This suggested failure mode hierarchy provides the means for automatic determination of dangerous/safe, non-critical failure modes (F0 categories). The suggested steps are:

1. The user must select one level 1 (F1) category.
2. This selection will then *automatically determine* whether the failure is "Dangerous", “Safe/Spurious” or "Non-critical".

3.4 Automatic determination of failure class (approach 3)

By combining the failure modes for each equipment group with the determined detection method, the failure class (DU, DD, Safe or non-critical) can be automatically determined. This is illustrated and summarized in Table 5.

Table 5: Automatic failure class determination based on detection method and failure mode

Failure mode (F1)	**Detection method**		
	Undetected		**Detected**
	1. Scheduled activity	**2. Unscheduled activity or event (demand or causal observation)**	**3. Diagnosed / immediately detected event**
SFI	DU	DU	DD
SF	S	S	S
NONC	NONC (Degraded / No-effect)	NONC (Degraded / No-effect)	NONC (Degraded / No-effect)

By further selection of failure mode at level 2, the non-critical failures can (for most equipment types) also be classified into degraded failures and no-effect failures (see for example Table 4 and Table 5-2 in [1]). This split can, together with the other input concerning failure mode, be used as automatic input to maintenance prioritization [1]. Figure 8 illustrates the potential for automatic determination of failure class as well as the resulting input to maintenance prioritization.

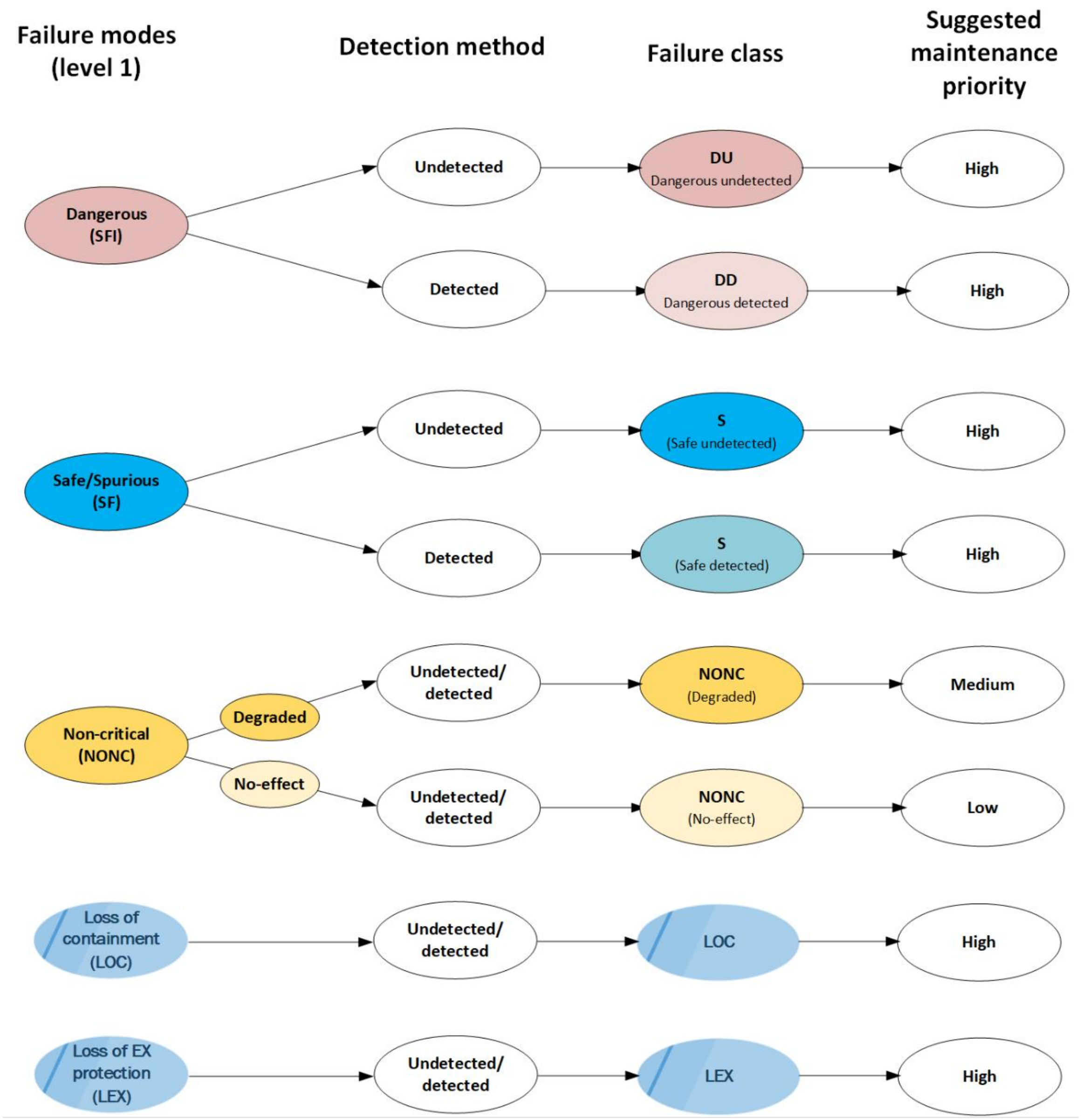


Figure 8: Automatic determination of failure class and maintenance priority (based on failure mode and detection method)

The suggested maintenance priorities assume that safety critical and potentially production critical (Safe) failures will be restored immediately (high priority). Degraded failures with a potential to develop into critical failures are given medium priority, whereas no-effect failures are given low priority. These suggested priorities can of course be manually overruled.

3.5 Challenges with automated reporting and determination of failure class

To achieve more automated and higher quality reporting of failure mode and detection method and determination of failure class, there are a number of conditions and prerequisites that must be present, some of which are discussed briefly below:

Trust in the data

Trust in the data is a paramount prerequisite to enable automated follow-up of data for safety equipment. For instance, it is of utmost importance that the correct data is provided, and that this data is complete with high quality. This implies that all failures, regardless of the detection method are registered against a specific tag. Furthermore, if the tag number is not correctly reported, it does not help to improve the classification approach. At the same time, use of logic conditions can reveal erroneous input, e.g. if a specific failure mode for a tag thought to be a shutdown valve is not available, this could indicate that the wrong tag has been chosen (or that the tag number is not correctly configured in the system).

It should also be ensured that use of the data and information cannot cause unintended consequences for production or other processes, for instance if the data sets and information are intentionally manipulated.

Standardization

Standardization is perhaps the single most important prerequisite for more automated reporting and for increased digitization in general. As illustrated in part 1, standardized hierarchies for failure modes and detection methods are a prerequisite for automatic determination of failure class.

For equipment failures, events, alarms, etc. to be machine readable there must be a common unambiguous language ("semantic interoperability") at the core, as well as a standardized format for data exchange. Furthermore, there must be a global system for identifying events, fail/pass criteria, and equipment (tagging, equipment classification) and all components must be described explicitly in relation to their functionality. Currently there are many different formats and definitions, which complicates the process and enforces the use of translation tables / mappings.

Other Challenges

It is important to keep in mind that some processes are not as easy to automate as could be expected. As a simple example, if you want to update a test interval based on demand rate, it will not always be straightforward for the system to distinguish between a functional test and a real demand. In addition, some systems, like PSD, have shutdown functions to avoid cascading, meaning that filtering must be performed to identify the actual demands.

Another example is the automatic determination of failure modes for shutdown valves. To be 100% certain about a failure to close (FTC), the feedback from the limit switch must be verified to be correct (which is not necessarily possible without a manual check in field).

Another challenge concerns access to information. The system operator must have access to all data and information regardless of which system or supplier is responsible or has delivered the solution, which is not always the case as per today. One example could be that a supplier often has access to more detailed data about their field equipment than the operator company does, often due to data or system limitations.

4 Automated entity annotations for notification text

4.1 Natural language processing (NLP)

To reduce the workload from manual failure class determination, natural language processing (NLP) is a potential candidate for automatised failure class determination from notifications. NLP is a technique for making human-generated texts or speech (e.g. Norwegian, English) comprehensible for a machine [16], as illustrated in Figure 9. Examples of NLP applications include machine translation, the auto spelling and grammar check in Microsoft word, information retrieval, and dialogue chatbots.

NLP research began in the 1950s for automatic machine translation from Russian into English [17]. One of the first NLP programs, ELIZA was developed in 1966, which was a chatbot demonstrating the dialogue between a psychotherapist (the program) and a patient (the user). According to [18], ELIZA was capable of identifying key words in the input text from the user, and generate output responses based on some rule set. Another example is a program called SHRDLU developed from 1968 to 1970. This program operated in a limited domain (i.e. a small simulated world of coloured blocks) and comprehended commands from the users given as natural language texts, by combining syntax analysis and reasoning about the domain world [19]. From the 1980s, machine learning methods have been deployed for enhancing the accuracy in NLP applications [20]. To date, NLP has been applied for information extraction in various text types such as clinical texts in health care systems [21] and legal precedent data [22].

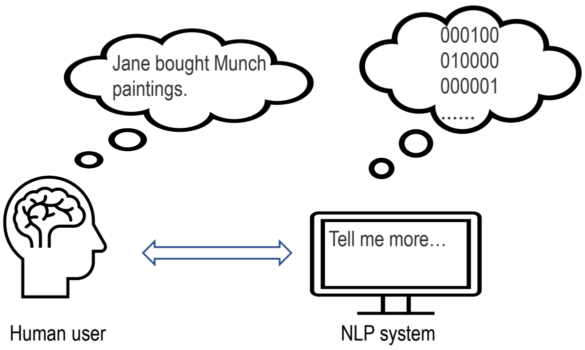


Figure 9: The role of NLP in analysing natural language text data

In the context of the APOS project, NLP has the potential to be applied for engineering texts like maintenance data and notifications. However, engineering text often contains domain-specific terminologies and abbreviations, which often calls for additional processing of the text to enable application of NLP techniques. For this reason, the term Technical Language Processing (TLP) has been suggested to refer to NLP applications for engineering texts [23].

4.2 Technical language processing (TLP)

TLP can be performed by using the common NLP pipeline that may be divided into two parts, pre-processing of raw text data and subsequent text analysis [24]. Pre-processing in NLP starts with text segmentation, a task of separating a text (corpus) into sentences. Each sentence is in turn decomposed into a set of words (tokens), and this task is called *tokenization*. Tokenization is usually followed by filtering out stop words which do not have substantial lexical meanings, for example, 'a, and, of, in, is, the'. Filtering of stop word, see Figure 10, is performed at an early stage to reduce the computational efforts and to focus on analysing more important words.

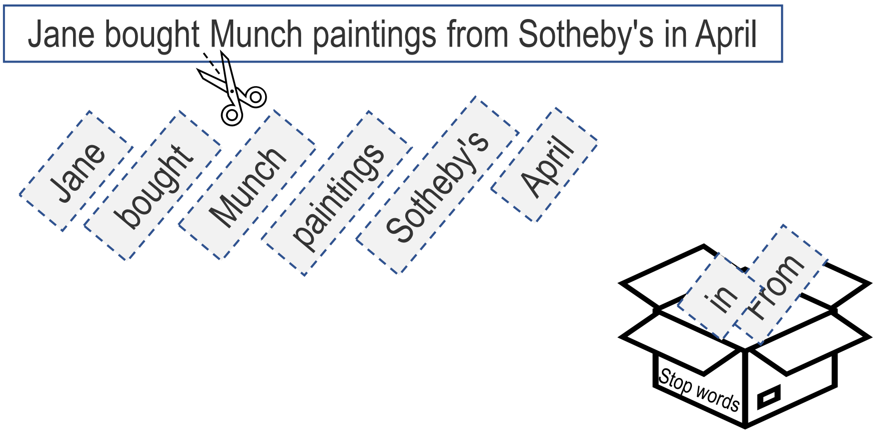
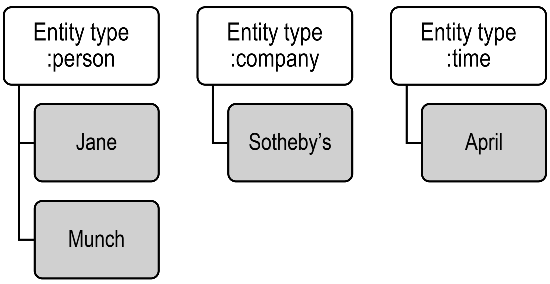


Figure 10: Illustration of tokenization in NLP and filtering out stop words, modified from Lane et al. [9]

Tokenization and stop word removal can be followed by text normalization, with the use of *stemming* or *lemmatizing*. Stemming is simply to cut off the commonly used participles like 'ed', 'ing', suffixes like 's', 'ful', 'ness'. For instance, the word 'paintings' can be stemmed into 'painting' and then to 'paint' which is the portion containing the meaning of this word. This means that stemming of this word makes it unnecessary for the machine to distinguish between 'paint' and 'paintings'. In this respect, stemming is used to reduce the size of the dataset and thereby enabling large-scale keyword search. Lemmatizing, on the other hand, is to look up lemmas or the root form of a word, for example, the word 'bought' is lemmatized into 'buy'. Lemmatization is potentially more accurate than stemming since it takes into account the meaning of a word, instead of simply cutting of the ending of a word [9]

A set of normalized words can be subject to various NLP tasks. One such task is named entity recognition (NER). A named entity is a word or a chunk of words that can be annotated or tagged with an entity type. NER can be understood as an approach to categorize information in a text by identifying 'named entities'. A named entity in NLP is a name for, e.g. a specific person, an organization or a location that occur frequently in natural languages. For example, a person name 'Jane' falls into the entity type 'Person', and a company name 'Sotheby's' can be categorized as entity type 'Company'. This is illustrated in Figure 11.

NER can be adopted for TLP use cases to categorize the information contained in engineering texts. To be able to use entity annotations for engineering cases, it will require that named entities (hereinafter entities) and entity types are defined for the specific use case. An index of entities and entity types, often referred to as a *dictionary*, may represent domain-specific resources that are not readily available in NLP and hence needs to be built based on expert knowledge.



SINTEF

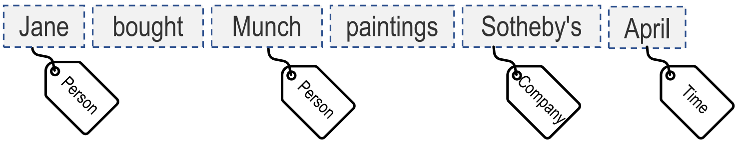


Figure 11: A simple example of NER for an example sentence.

For our purpose entity annotation or tagging of maintenance data is of special interest. Text annotation for maintenance data can be used to identify information of particular interest that have previously been "hidden" or unknown to the user (failure mode, detection method, etc.). Moreover, the index of entities and aliases can be used to tag a larger dataset. The information gained from the analysis of the tagged data can be used in important maintenance decisions, such as which equipment to prioritize [25]. Woods et al., [26] propose a list of maintenance activity terms with associated hierarchies used in maintenance work description texts based on ISO 14224 and ISO 15926-4. For example, the entity type 'replace' will have corresponding entities like 'change', 'reinstall' and 'swap', and the entity type 'inspect' may correspond to 'check', 'test', and 'visual inspection'.

Prior to entity annotation, it is important to pre-process raw text, which can be done by using the existing NLP tools or TLP resources. For instance, tokenization and removal of stop words may be executed using NLP tokenizers. On the other hand, stemming or lemmatization may require extra TLP sources to process infrequently occurring words in natural language and technical terminologies. Such words are sometimes written in abbreviations and with misspelling, for example, 'hyd', 'hydraulic', 'hydaulic' for the word 'hydraulic' [24]

4.3 A case study on manual entity annotation of notifications

Entity annotation for failure modes was briefly tested in a PDS workshop where PDS members being familiar with failure class selection were using an open access annotation tool. The main objective was to explore the potential for automated failure class determination from notifications using the taxonomy of failure modes suggested in the report "Guidelines for standardised failure reporting and classification of safety equipment failures in the petroleum industry" [1]. The dataset applied in this study was an extraction of 80 notifications for ESD valves obtained from three different oil and gas facilities in Norway. The short text field of the notifications were mainly used for the annotation task, and the length of the short text fields were 1-2 sentences and described the potential failure in a rather concise way.

The annotation was manually done by experts/expert group, using the tool RedCoat [27]. There are several text annotation tools available, each having their strengths and weaknesses (see Table 1 in [27]). Ottermo et al. (2021) proposed the following criteria for selecting suitable tools for text annotations for maintenance notifications, and found that RedCoat was the most suitable one for the task:

- Suitable tool with respect to technical language processing.
- User friendly, i.e., easy to annotate – particularly for persons not familiar with the tool.
- Possibility to establish a taxonomy or hierarchy of entities without any restrictions on numbers and levels, and possibility to adjust the hierarchy during the annotation process.
- Tailored for collaborative annotation.
- Optimized for short texts.

Prior to the annotation workshop, a simple user manual was developed to guide the participants during their annotation. None of the participants had any experience with TLP prior to the workshop. In the beginning of the workshop a brief introduction to TLP, the tool, and the annotation task was given. Then, a total of 20 participants familiar with safety critical equipment performed individual annotation of the notifications with respect to failure modes. All participants were asked to annotate the same 25 notifications. The participants were instructed to annotate the dataset with one or more failure modes, meaning that all possible failure modes could be selected for each notification.

In addition to the participants that performed individual annotation, a co-operative annotation session in a larger group of system experts was facilitated. About ten of the notifications were reviewed and discussed. The discussion among the participants during the annotation session and the subsequent plenum discussion, revealed several interesting issues regarding the annotation.

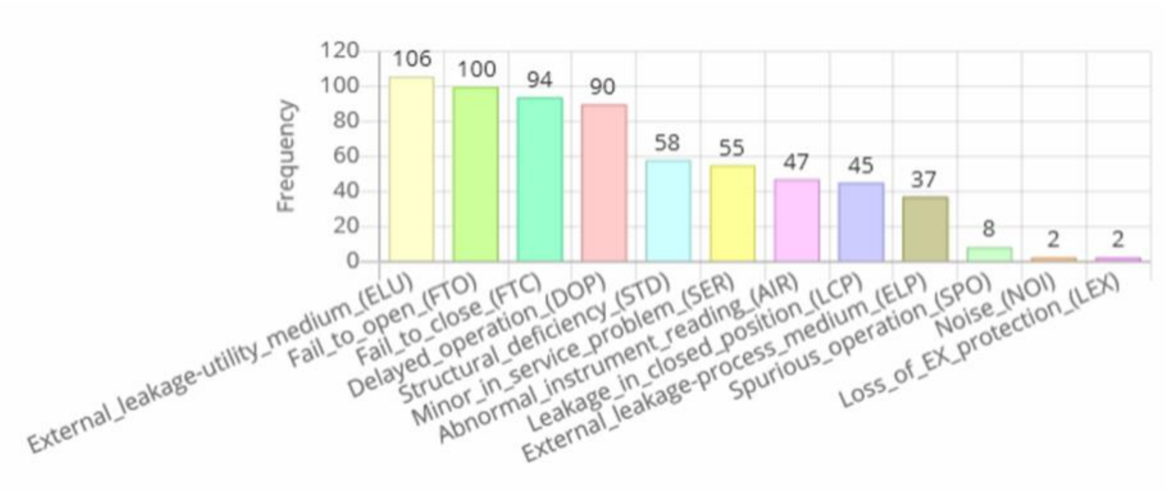


Figure 12: The number of annotations for each failure mode from the workshop

As a result of the workshop, the entity frequencies of the failure modes were obtained as illustrated in Figure 12. The frequency of a failure mode equals the total number of annotations in the 25 notifications among all 20 participants. All 12 failure modes of the established taxonomy had been used at least once. The most frequent failure modes were external leakage of utility medium (ELU), fail to open (FTO), fail to close (FTC), and delayed operation (DOP). This result may be interpreted in several ways:

1. ELU, FTO, FTC, and DOP are (the most) common failure modes for shutdown valves. Of course, this assumption needs to be further substantiated based on a larger dataset than only 25 notifications.
2. ELU, FTO, FTC, and DOP occur most frequently since these failure modes are often annotated together with other failure modes, for instance, FTO and FTC have often been selected for the same text due to uncertainty.
3. ELU, FTO, FTC, and DOP are the most familiar failure modes to the annotators, either with respect to criticality (as FTC, DOP, and FTO are all dangerous failures for shutdown or blowdown valves) or with respect to occurrence (for example ELU is a well-known problem for valves).

The agreements for entity typing of each notification was between 13% and 68 %, and this implies that manual entity typing can still be subject to bias of the annotators. There are three main reasons for this:

First, annotation was done in a non-uniform manner for some notifications. In other words, individual annotators had their own preference in the number of the words that they used for entity tagging. Some annotator used two key word(s), while some annotators used a whole sentence. For instance, for a notification tagged with DOP, some annotated only the word ‘Stengetid’ (‘Closing time’ in English), while others annotated the phrase ‘Alarm paa stengetid’ (‘Alarm on closing time’ in English).

Second, the understanding of the free text can vary between the annotators. For instance, 'Alarm paa stengetid' ('Alarm on closing time' in English) can be interpretated in three different ways.

a) The shutdown valve not closing 100% and thereby not giving a closing alarm, which implies FTC
b) The valve closing 100% but not within the response time requirement (also giving an alarm), which implies DOP
c) The valve has closed spuriously, and this spurious closure is revealed by an alarm, which implies SPO.

Third, in many cases, multiple failure modes could have been selected. However, some annotators ended up choosing the most critical failure mode, while some may have selected the first failure mode that the annotator associated with the text.

The main conclusion from the workshop was that the manual entity typing is a good exercise to involve experts in failure classifications but is still subject to bias and variances in how individual annotators perform entity typing. Moreover, it is time-consuming to analyse a large number of notifications. To overcome some of the limitations regarding manual annotation of notifications, an approach for automated entity annotation and failure class determination is suggested.

4.4 Proposed approach for automated annotation

To overcome the limitation regarding manual annotation of notifications, an approach for automated entity annotation and failure class determination is suggested. The main objectives of automated annotation for notifications are to:
1) Extract the minimum information needed for failure class determination in an iterative and reproducible way.
2) Classify a larger number of notifications in a shorter timescale.

As a starting point, we utilized the RedCoat tool with its automated annotation feature, which requires establishing the entity type hierarchy and creating an entity dictionary for entity annotation. In our case, the entity hierarchy is rather simple, as shown in Figure 13. RedCoat requires that the dictionary contains at least one entity for each of entity types registered in the hierarchy.

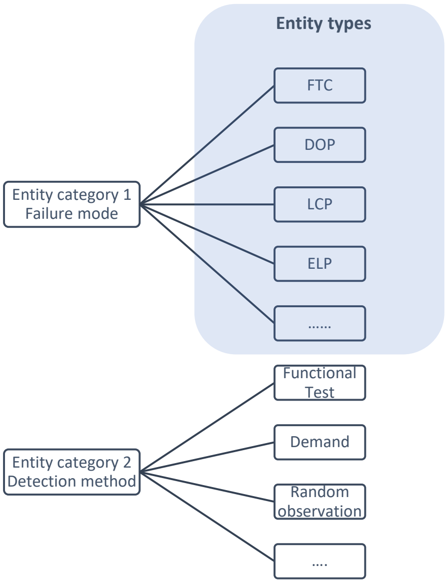


Figure 13: Entity hierarchy for notification data

Although Redcoat is a user friendly and open access tool, it was for practical reasons (and some limitations in Redcoat) decided to a built-in Excel algorithm for automated annotation. This requires five main steps. Figure 14 illustrates the suggested TLP approach applied for the ESD valve case.

1) Pre-process the (raw) text
2) Establish dictionaries for the failure modes and detection modes for the given equipment
3) Use the dictionaries for entity type tagging
4) Classify each of the tagged entity
5) Classify the failure

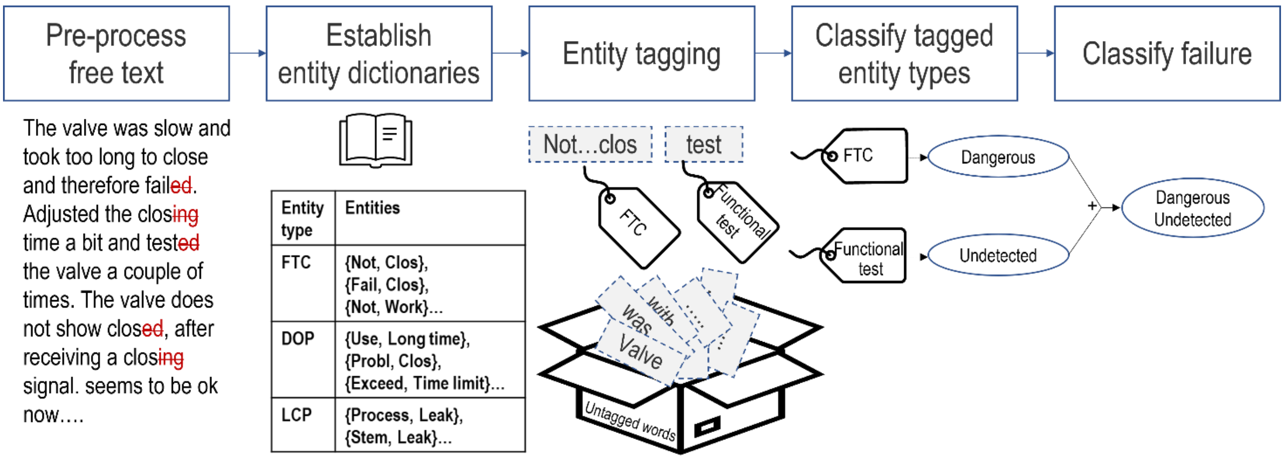


Figure 14: A simplified illustration of suggested approach applied for ESD

4.4.1 Proposed approach

The applied dataset was compiled from exports of registered notifications for ESD valves. For each specific plant, the notifications for a further specified time period (e.g. one year) were listed in an individual excel worksheet. Depending on the plant, the format of the notifications can vary slightly with different sets of data fields. An example of some typical fields required for a notification was presented in Table 2. Despite the slight differences, a notification usually includes a set of data fields, such as notification number, tag number, location (of the equipment), detection method, failure mechanism, failure cause, failure mode, and free text. Some plants have both short text and long text fields. Data fields like 'Tag' and 'Object part code' may be generated automatically, while fields like 'Cause code', 'Problem code' can be a dropdown menu.

Two data fields necessary for failure class determination are failure mode and detection method. There is usually a dropdown list for failure mode from which the technician can choose an option that will match the observed failure. However, often the options 'other' or 'unknown' are chosen or the field may even be left as blank. Moreover, detection method fields were left blank in many of the notifications. Instead, the description about the equipment status and detection methods were included in the free text. Example of a notification free text for an ESD valve written in Norwegian is:

EV-ventil åpner ikke. *24.10.2011 19:07:16 Mr. H. * Skulle åpne EV-xx-xxxx på væskefanger BB for trykkavlastning, men den ville ikke åpne. Utetekniker sjekket lufttrykk, og det lå på 3,5 bar. Justerte opp til 5 bar, men ventil åpnet seg ikke da heller. Ventil bevegde seg heller ikke ute i felt og vi fikk svikt på ventilen. *25.10.2011 06:21:10 Mr. S. 178118. *Endret Functional location til EV.SD002 Depressurisation System. * 25.10.2011 06:43:31 Mr. S . * Så lenge væskefanger BB er avstengt i forbindelse med vaskejobb, er ikke denne kritisk. Væskefanger er trykkavlastet i den sammenheng. Prioritering satt i forhold til kostnad forbundet med at vi ikke får tilbakestilt væskefanger. Det må feilsøkes på aktuator evnt se på difftrykk og hvirken innvirkning det kan ha på ventilen. * 25.10.2011 10:41:07 Mr. A. * Ventilen er testkjørt sammen med uteoperatør Mr. D i dag og ventilen går fint ute i felt. Ventilen kommer i svikt når den får åpnesignal, antakeligvis trøbbel med tilbakemelding på åpen posisjon.* 25.10.2011 12:06:46 Mr. H * Denne notifikasjonen avsluttes da feilen ikke ligger i ventilen. Ventilen går nå som normalt men har en feil på tilbakemelding åpen pos. Dette er meldt i ny notifikasjon xxxxxxx.

As seen from the example free text above, a long text field can generally include reporting of maintenance activity (e.g. type of testing, repair actions, repair history, etc.). Moreover, detailed description of failures or technical degradations are provided, despite that the text is unstructured, and often contains misspellings and abbreviations. It is therefore reasonable to extract key information needed for failure class determination, rather than reading through the whole text. This key information can then be used to suggest, supplement or quality check information in the 'failure mode' and 'detection method' data fields. In light of this, a stepwise approach is suggested for automated entity annotation of free texts, where built-in functions in Excel is proposed for tagging of entities. The tagged entities are then used to classify failures.

Step 1 Pre-process the notification text
Many of the notification texts for ESD valves include sentences describing the valve movement. These sentences are essential for understanding the valve functionality, and hence for classifying failures. An example of such a sentence is 'Ventilen var treg og brukte for lang tid på lukking og gikk derfor i feil' (English: The valve was stuck and used too long time to close and open, and thus failed). The word 'lukking' (English: closing) in this sentence is a variation of the basic form 'lukke' (English: close) with the added suffix part 'ing'. Depending on the writing style of a technician, the verb 'lukke' can be written in different forms, e.g., 'lukker' (English: is closing), 'lukket' (English: closed) and 'har lukket' (English: have closed) with different tenses. In such cases, stemming can be applied to cut off the word endings like 'ing', 'er', 'et' and

keeping 'lukk' (English: clos) (see Section 4.2). In addition, stemming can be applied for nouns, e.g., lekkasje, lekasje, lekker → lekk (English: Leakage, leaking → leak).

The stemming is performed by using search-replace in Excel, instead of utilizing existing NLP tools. The main reason for is that we suggest to perform the stemming only for specific key words (e.g., words related to valve failure) and to disregard or remove the rest of the words in the next step. For this reason, a review of the free texts was required to list the words whose endings can be removed, as a preparation for stemming. This review revealed that free texts were written in Norwegian, and very few of them were written in other languages (i.e., English, Swedish, Danish), and that there were only a limited set of vocabulary used for reports related to maintenance of ESD valves. In light of this, it was deemed feasible to list the key words to be searched and replaced, rather than trying to connect existing Python tools (e.g., Snowball) [28] to Excel, for supporting stemming of Norwegian texts.

Step 2 Establish the dictionary for failure modes and detection methods
The free text pre-processed in Step 1 should ideally be annotated or tagged with two entity types, i.e., one failure mode and one detection method. For this reason, we suggest using two dictionaries, one for failure mode and one for detection mode, such that each dictionary is used as an index for finding entities, see

Figure 15.

The failure mode dictionary will list a set of entity types (failure modes) for the given equipment and the associated entities (key words) for each of these failure modes, as exemplified in Table 6. For instance, the entity type FTC for ESD valve will have entities like 'Not closing', 'Fail to close', 'Problem with closing'. In the same way, entity types and entities for detection modes can be listed. For example, 'partial stroke testing' and 'periodic function test' can be identified as entities for the entity type 'functional test' for ESD valve.

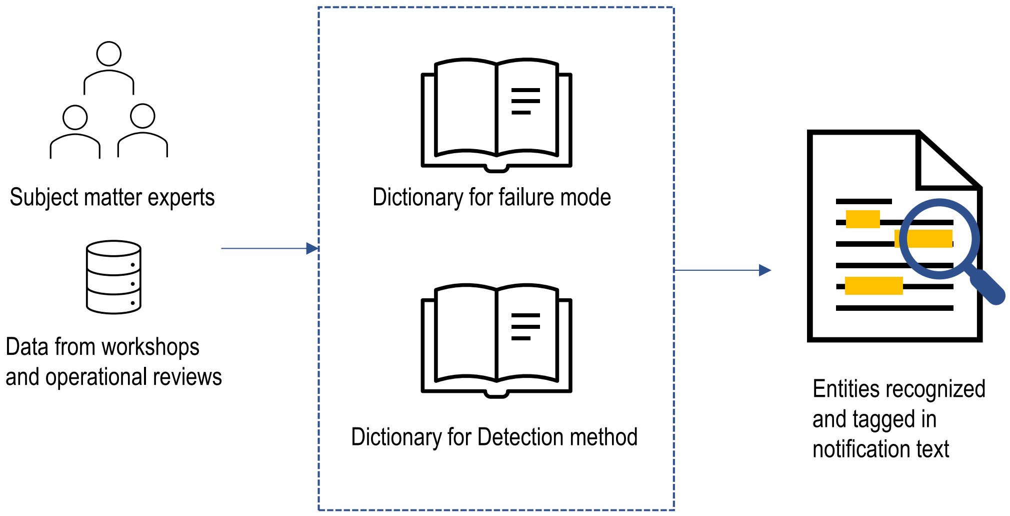


Figure 15: Entity annotations for notification text.

Table 6: Example failure mode dictionary (incomplete)

Entity type	Entity (examples)
FTC	Norwegian: Lukker ikke, Kjører ikke, Går ikke, Virker ikke, Problem med lukking, Vanskelig å lukke, Ventil sitter fast
DOP	Lang gangtid, Lang stroketid, Problem med lukking, treig å stenge, Langsom bevegelse, Hastighet på ventil økes, For lang tid, Gangtidsfeil
LCP	Lekkasje i stengt posisjon, Lekker gjennom ventilen, ventil ikke tett, lekker i lukket posisjon
SPO	Ventil stengte tilfeldig, stengte uten signal, ventil trippet, stengte plutselig
FTO	Åpner ikke, Åpnet ikke, kan ikke resettes, Treg å åpne, Kjører ikke, Går ikke, problem med åpning, sitter fast lukket

Step 3 (Post) processing the dictionaries

This step is necessary to be able to identify key words and phrases used for entity recognition (i.e., search of key words). An example of search phrase is 'fail to close' for ESV valve, which may often be written in different ways like 'fail to close', 'failure with close', and 'fail in the valve closure'. This means that the number of entities to be included in the dictionary will be high, if we include all variations. For this reason, we use tokenization, stemming and removal of stop word techniques to process the dictionaries made in Step 2. For example, 'failed to open' can be tokenized into {failed, to, open} which can be reduced to {failed, open}, then to {fail, open}.

Moreover, we quickly realized that description for many failure modes of ESD valve include negative words like "not", "problem", and "fail", together with a verb describing the function of the equipment like "open", "close", "go". The negative word can be placed both before and after the verb in the free text, e.g., "failed to close" or "closing failed", and this results in many possible entity combinations. For the purpose of limiting the entity list, it may therefore be reasonable to use a 2-step entity recognition process for many of the failure modes. The list of entities for entity type 'FTC' is shown in Table 7.

Table 7: List of entities (in Norwegian) for entity type 'FTC'.

Entity type	**FTC**	
	Step 1 (Negative word)	Step 2 (Verb or Noun)
Entities	Ikk	lukk
	Feil	naar
	vansk	steng
	treg	virk
	lang	roer
	mellom	kjoe
	skad	gaa
	blokk	gikk
	gnisning	bytt
	tett	bruk
	fast	staa

	….	sitt
		stengetid
		gangtid
		tilbakemeld
		just
		…

Step 4: Annotate notifications with entity types
A simple algorithm using standard Excel functions has been developed to annotate notifications based on the indexes, as illustrated in Figure 16.

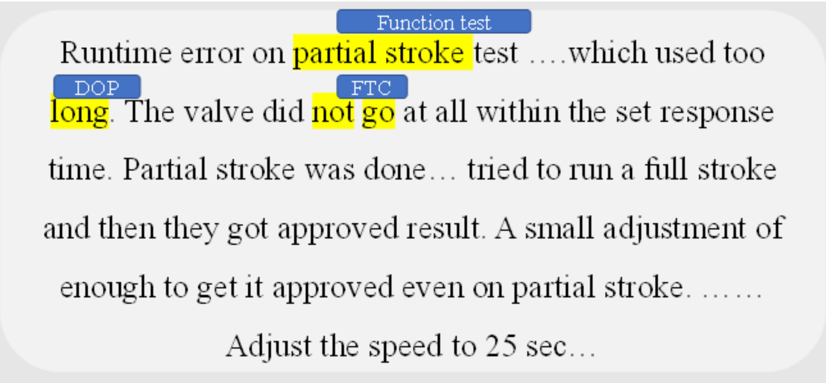


Figure 16: Entity annotation with the defined entity types for failure modes and detection methods

The Excel formula used for entity annotation for each failure mode is:
IFERROR(MATCH(TRUE;ISNUMBER(SEARCH('Dictionary - Failure mode ESD'!$B$3:$B$18;C15));"0");0)

We used the SEARCH function to read a notification text and compare the text with each of the entities listed for each entity type (e.g. FTC). If the first entity in the list is found in the notification text, it will return the number of the characters in the text before this entity is found. Otherwise it will not return any value [29]. Next, the SERACH function will check if the second entity is found in the text, and the same way until the last entity in the list is checked. Then, the ISNUMBER function is used to replace the numbers with 'TRUE', and to replace no value with 'FALSE'. Then, we combine the MATCH function and IFERROR function to check if there is 'TRUE' values in a range of cells, and returns the relative position (i.e. first, second, third) of that item in the range [30]. If there is only 'FALSE' in the range, the returned valued will be 0. This process is repeated until each of the entity type for the failure is visited. Next, detection methods are annotated by using the same formula but using the dictionary for the detection methods.

Step 5: Annotate entity types and classify failures using the annotated entity types
In this step, the notifications are annotated with entity types. The tagged entity types are used for failure class determination, as illustrated in Figure 17.

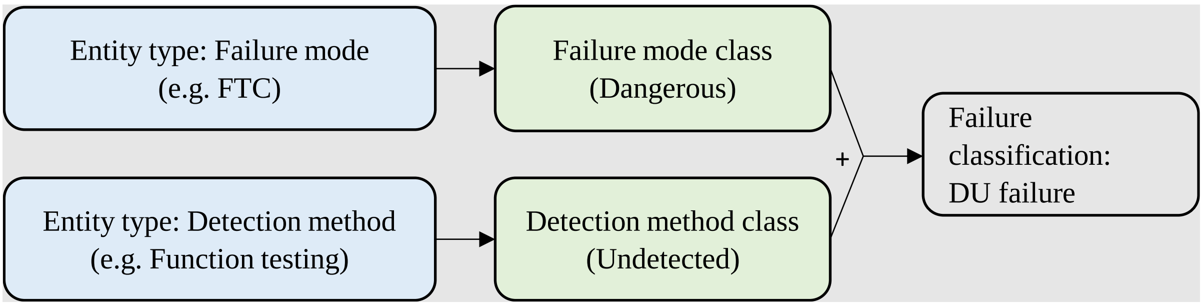


Figure 17: Failure class determination by using the annotated entity types

The IF function in Excel is used to classify the entities in step 3 with the corresponding entity type classified according to Table 8, (e.g. FTC is classified as 'dangerous'). Degraded failure and no-effect failure are both classified as 'non-critical failure' as was shown in Table 8.

It should be noted that two different entity types can have common entities. For instance, the entity types FTC and FTO have common entities like 'not moving' (In Norwegian: går ikke). This means that a free text that contains the phrase 'not moving' will be annotated with two entity types, FTC and FTO. In such cases, the failure mode class will be 'dangerous' or 'safe'. When this occurs, the most conservative failure class is chosen, so 'dangerous' will be chosen over 'safe'. In other words, in case of multiple failure mode classes, one failure mode class is chosen according to the severity of the failure mode, as shown in Figure 18.

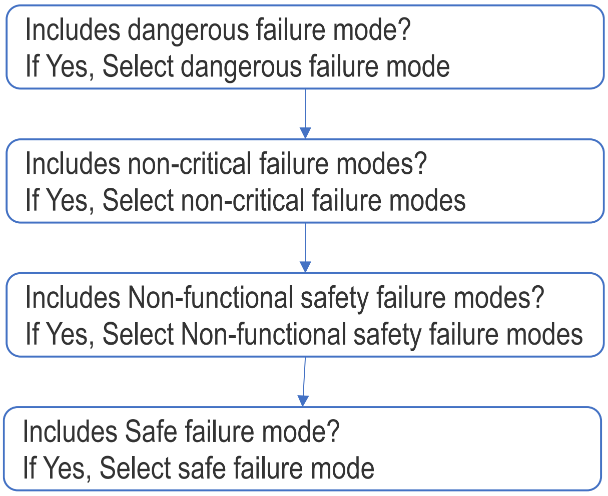


Figure 18: Failure class determination based on annotated entity types

As for failure mode, detection method entity types are categorized with detection method class, as was shown in Table 3. Once the failure mode class and detection method class are assigned, they are combined to obtain the failure class. Possible failure classes are then dangerous undetected (DU), dangerous detected (DD), safe, non-critical (LOC, LEX).

Table 8: Failure mode hierarchy for shutdown valves

F0	Failure mode (level 1)	Failure mode (level 2)
Dangerous failures	Dangerous failure	Fail to close (FTC)
		Delayed operation (DOP) * 1)

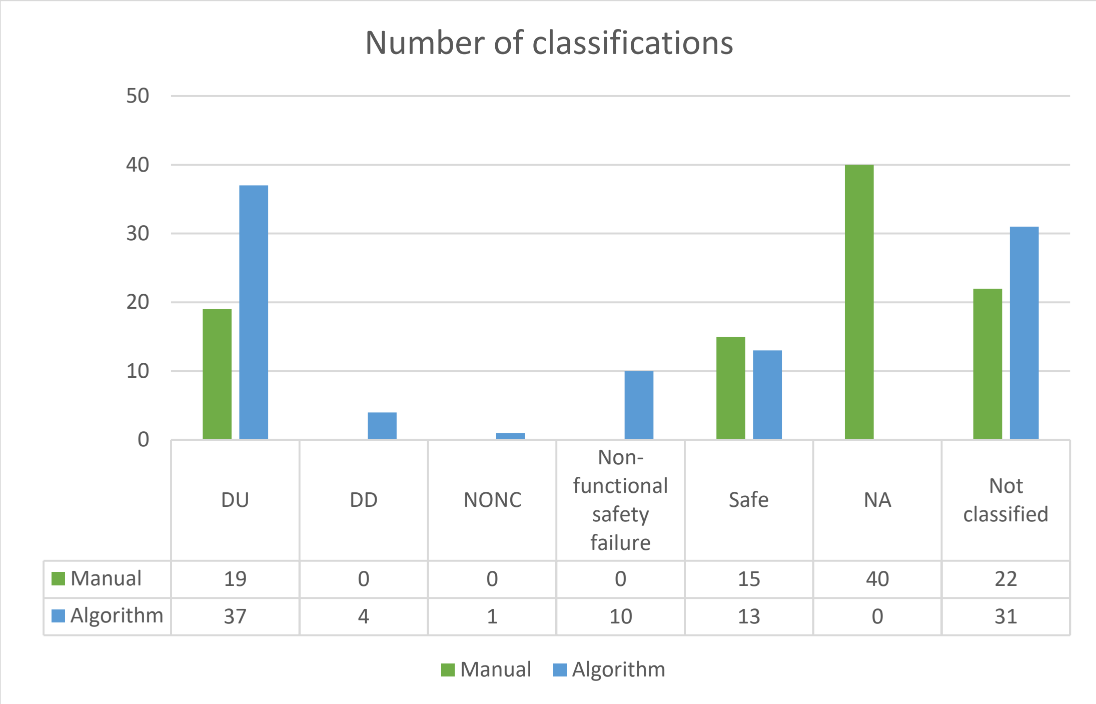
F0	Failure mode (level 1)	Failure mode (level 2)
		Leakage in closed position (LCP) * 2)
Safe failures	Safe failure	Fail to open (FTO)
	Spurious failure	Spurious operation (SPO)
Non-critical (NONC) failures	Degraded failure	DOP * 1)
		LCP * 2)
		Structural deficiency (STD)
	No-effect failure	Noise (NOI)
		Abnormal instrument reading (AIR)
		Minor in-service problems (SER)
Non-functional-safety failures	Loss of containment	External leakage – utility medium (ELU)
		External leakage – process medium (ELP)
	Loss of explosion (EX) protection	Loss of EX protection (LEX)

1) Only relevant if response time requirement given.
2) Only relevant if internal leakage requirement given.
* the same failure mode on level 2 is relevant for more than one failure mode on level 1 (e.g. an internal valve leakage above versus below the specified acceptance criterion).

4.4.2 Results from ESD valve case study

The suggested approach was applied to two cases. The first dataset included 96 notifications from an onshore plant. Nineteen of these notifications were classified as DU during the manual operational review (i.e. the answers/solution was given). For the rest of the notifications, 15 were classified as safe, and 40 notifications as non-applicable (NA) and 22 were not classified.

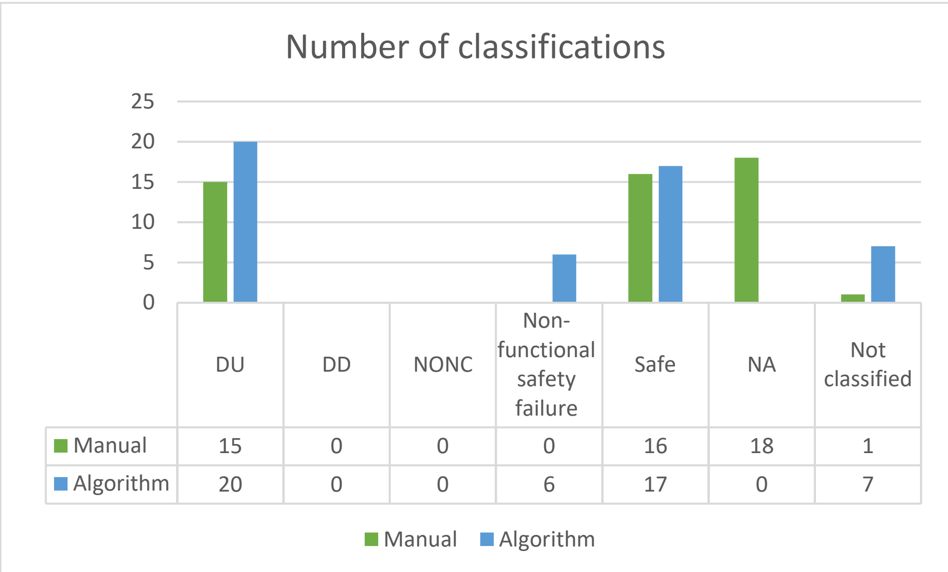
Using the algorithm (i.e. automated failure class determination) resulted in 37 DU notifications, and we have checked if the actual 19 DU failures were included in these 37 DU failures. Fifteen out of these actual 19 DU failures were also classified as DU failures by use of the algorithm. The four mismatched notifications included: Three of the notifications were written in English, and since the dictionary was written in Norwegian no failure class was determined. The final mismatched notification was classified as a DD failure by the algorithm. The notification was annotated with FTC, FTO, SPO and SER, and conservatively classified with the most dangerous failure mode (FTC). However, the notification was annotated with ‘Detected’ due to the entity ‘Alarm’ that was included in the free text. However, the failure was in fact detected during a ‘Demand’. The number of classified failures for the 96 notifications dataset is presented in Figure 19.



	DU	DD	NONC	Non-functional safety failure	Safe	NA	Not classified
Manual	19	0	0	0	15	40	22
Algorithm	37	4	1	10	13	0	31

Figure 19: Number of classified failures for 96 notifications from an onshore plant

The second dataset include 50 notifications for ESD valves obtained solely from functional tests. The same dictionary that was established based on the first dataset (case 1) was used also for this case. The algorithm classified 20 notifications as DUs while 15 notifications were classified as DU during the operational review. By comparison, 13 DU failures matched. For the two unmatched notifications, the first notification did not include any free text, and was not possible to classify. The second notification was annotated as FTO and hence classified as safe failure, while it was classified as S/DU in the manual review. The number of classified failures for this dataset is summarized in Figure 20.



	DU	DD	NONC	Non-functional safety failure	Safe	NA	Not classified
Manual	15	0	0	0	16	18	1
Algorithm	20	0	0	6	17	0	7

Figure 20: Number of classified failures for 50 notifications from an offshore plant

4.4.3 Discussion

The two case studies show a fairly good match in the DU classification between the automated approach and the results from the operational review. However, the results from the two cases imply that the algorithm is too conservative. For the first case, the automated approach gives about twice as many numbers of DU failures (37 DUs) as the actual DUs identified from the manual (operational) review (19 DUs). For the second dataset, the algorithm gives an additional five DUs as compared to the manual review. There are several reasons for this. First, the approach will only base the failure class determination on the detected key words (entities) in a notification text, rather than full sentences. For instance, the algorithm tagged the text below with FTC, DOP, and FTO while the operational review classified this notification as Safe. For FTC, the automated approach detected the words 'ikke' ('Not' in English)' and 'stengte' (closed in English)'. For DOP the detected words are 'problem' ('Problem' in English)' and 'stengte' (closed in English)'. For FTO the detected words were 'problem' and 'gaa' ('Move' in English). However, if we read the full text, it says 'Ventil kjører fullstroke på partialstrok' ('Valve runs full stroke on partial stroke' in English), which the algorithm did not comprehend. The second reason is that the algorithm is designed to choose the most critical failure mode class with respect to the specified safety function, i.e., in case multiple failure modes are annotated, the most dangerous failure mode(s) will be used for the failure class determination. For example, for the text below, failure mode is first annotated with FTC, DOP, and FTO. The algorithm will choose FTC or DOP as dangerous whereas FTO will be discarded. FTC is then selected among the two. Furthermore, the algorithm does not identify if words are in the same sentence, or a combination of words are used to provide some meaning. For example, the algorithm detected 'stengte' in one sentence and 'problem' in another sentence in the text below, and then tagged the text with 'FTC', according to the combination of the two words. However, the 'stengte' in the sentence is followed by 'helt' ('completely' in English), which does not indicate any failure modes, but rather that the valve fully closed. On the other hand, the word 'problem' was part of the sentence 'problemet her er at ventilen stenger for raskt og ikke partial stroke tidene' (English: 'problem here is that the valve closes too quickly and not the partial stroke time'), indicating that there is a problem with the valve closing too quickly rather than the valve not closing properly. Such an issue arises from removing stop words like 'with', 'in' in the dictionary for reducing the size of the entity list.

> NO.xxx.EV -16-2001 Ventil kjører fullstroke på partialstrok
> * 09.12.2011 05:22:09 Ms. H* Under A1 partial stroke test så stengte ventilen helt når vi kjørte delvis slag.* Fikk et lite trykkfall på 2barg etter ventilen.* ABB bør se på saken!
> ** Stopper resten av partialstroke testen til de får sett på dette. * EV-27-4912 og EV-27-4913 står igjen på listen. Tar ikke sjansen på å gå på samme smellen på utløpsventilen.
> * 09.12.2011 07:53:08 Mr. F * Kompenserende tiltak er å stoppe partial stroke test inntil dette er sjekket og avklart. * 09.12.2011 13:34:13 Mr. F * Jeg har sett litt på denne Notifikasjonen . * Gangtiden på denne ventilen er etter 28/6 blitt vesentlig raskere (rundt10s mot tidligere ca 30s)
> * Kan ikke finne noe i SAP på at gangtiden er justert, men ser det er gjort en jobb på å tette hydraulikk lekkasje på fra aktuator 28-29/6 2011** Jeg mener at problemet her er at ventilen stenger for raskt og ikke Partial Stroke tidene.* PSD Partial Stroke tid EV-16-2001 PPST er i dag satt til 15s
> * ESD Partial Stroke tid EV_16_2001_ET OUT er i dag satt til 15s

It should be noted that the algorithm did not classify some of the notifications at all. One reason being that focus has been on establishing entity lists for dangerous failures, while less attention has been given to the entity lists for Safe, NONC and non-functional failures. For example, the key word 'tett filter (clogged filter in English)' could have been an entity for the failure mode 'SER', which was however not included in the dictionary. Hence, if the dictionary for non-dangerous failures is refined, the algorithm will become more precise but should still (when more than one failure mode is relevant) select the most dangerous failure modes over others to ensure that these are identified. Hence, the number of identified DU failures will probably still be higher for the automated algorithm compared with the manual review.

Tett filter på IV-skap * 17.07.2011 13:42:13 Mr.E. * Filter har gått tett i IV-skap på ventil. Pop up indikator. (Dette IV-skap har 2 filter, det ene har gått tett) * 18.07.2011 06:35:01 Mr. S. * Ikke brudd på PS. Ingen kompenserende tiltak nødvendig.

Several other refinements to make the algorithm more intelligent are possible. For instance, the text below is not tagged with any entity for failure mode, because the word ‘close’ does not appear in combination with negative words like ‘not’ or ‘problem’ and is thus not classified at all. On the other hand, the failure is classified as ‘safe’ according to the operational review. This means that a better understanding of the meaning of the text is necessary to precisely classify the notification in some cases, implying that entity tagging may not be sufficient on its own. Also, an autocorrect of spelling errors would further improve use of the algorithm (or reduce the size of the dictionary).

Ventil stengte på 3 sekunder, skal bruke 16 sekunder. Brenngass til brennerbom, plassert rett ved trappa oppe på P14. 24.01.10 Har justert gangtid til 16-sek.

5 Combining automatic parameter determination with TLP

In Chapter 3 of this report, we explored the potential for more user aided and/or automatised pre-filling of notifications as well as automated determination of failure class. However, automated determination of failure class requires that the failure modes and detection method registered in the notifications are correct. As discussed in Section 2.3, this is not always the case as per today. This may have various causes, including too many (often abbreviated) failure mode alternatives. It has also been pointed out that selecting the correct detection method category can be difficult, and that the most relevant information about failure mode, detection method and failure cause, is often found in the free text field(s) in the notifications. Therefore, chapter 4 investigated how technical language processing (TLP) can be used to automatically extract failure mode and detection method from free text. In this chapter, we discuss some possibilities for combining the methods described in chapters 3 and 4.

5.1 Approach 1: User guidance on selection of failure mode and detection method

Several ways of displaying help-text to aid the maintenance personnel when selecting failure mode and detection method manually have been described earlier in this report. There are a couple of ways in which such user guidance can be further improved by using TLP techniques:

1. Online TLP recognition to check if the free text contains information about failure mode or detection method. This could provide both online guidance and quality assurance for the operator. Examples of such online guiding questions could be:
    a. You have selected the failure mode Fail to close (FTC) but based on what you wrote in the free text field you could consider Fail to open (FTO). Please check match between free text and selected failure mode!
    b. You haven't selected any failure mode but based on what you wrote in the free text field you could consider the failure mode Fail to close (FTC).
2. Online LP recognition to check if the free text contains information that suggest that the failure has been reported on the wrong tag. On example could be:
    a. Based on the long text you have written we suggest that there is a failure on the solenoid, but the notification is written for the main valve. Are you sure you have selected the right tag number?

5.2 Approach 2: Automatic determination of failure mode and detection method

In chapter 3 we also described some possibilities for collecting data and information from relevant OT systems such as SAS, IMS, CMMS that could be used for automatic determination of failure mode and failure detection.

In such cases, there could be a possibility to use online text processing to verify that the signals that are automatically fed into the system are correct and vice versa:

1. One example could be to verify the information from an Automatic Shutdown Report (ARS). E.g. from the ASR it is not known whether the valve closed or not, but information from the free text implies a failure on the limit switch.
2. And opposite: You have chosen the failure mode DOP, but from the SAS information it appears that the travel time was in line with the specified time

5.3 Approach 3: Automatic determination of failure class

It could also be possible to use a combination of TLP text recognition and pre-filled failure mode and detection method to evaluate the confidence in the failure reporting/failure class determination

1. For instance, if there is strong correlation between the long text and one or more of the reported failure modes or detection methods, it could be possible to grade how confident the system is that the correct failure mode/detection method/failure class have been identified, e.g. by using some kind of colour coding (as for the weather forecast). For instance, if the TLP text recognition suggest the failure class DU, and the automatically determined failure mode and detection method also indicate a DU, the confidence that the right failure class has been determined is high.

6 Conclusions and further work

In this report we have investigated possibilities for automated reporting of failure mode and detection method and failure class determination to reduce manual effort and subjectivity to ensure high-quality data. The main contributions from the work include proposals for how to:

a) Include pre-defined help-texts for improved manual failure reporting.
b) Automate determination of failure mode and detection method based on information from additional systems such as SAS, etc.
c) Automate failure class determination (DU, DD, etc.) based on standardized failure mode and detection method hierarchies.
d) Use of technical language processing (TLP) methods to extract key information for failure class determination.

The work indicates that there may be a high potential for increasing the quality and efficiency of failure reporting and failure class determination. However, there are a number of conditions and prerequisites that must be present to ensure a trustworthy process. For points a)-c) above, standardized definitions and/or hierarchies for failure modes and detection methods must be available. Furthermore, system information, such as equipment failures, events, alarms, etc. must be made machine readable, which requires a common unambiguous language ("semantic interoperability"). Semantic interoperability is further explored in the APOS report "Information model for functional safety" [4] and "Guidelines for standardised failure reporting and classification of safety equipment failures in the petroleum industry" [1].

For point d) above, the work indicates that TLP of free text fields can be a useful supplement to manual failure class determination from notifications. The case study showed that TLP can detect most DU-failures, however, the algorithm needs to be further improved to make it less conservative (too many non DUs are currently classified as DU). The case study revealed some examples where TLP failed to annotate and classify the text of the notifications. The reason was that the free text did not include any of the entities in the indexes. Therefore, it is necessary to *continuously* update the indexes and to further refine the algorithm and entity indexes. By resolving these issues, TLP may be a powerful tool for more consistent failure class determination and improved quality assurance of the manual failure class determination.

More case studies or demonstrators are recommended to document the cost-benefit of the approaches discussed in this report.

7 References

1. Hauge S, Håbrekke S, Lundteigen MA, Lee S, Ottermo MV. Guidelines for standardised failure reporting and classification of safety equipment failures in the petroleum industry, Ed. 1 (open version) (APOS H1). 2023. Report No.: SINTEF Report 2023:00108.

2. Håbrekke S, Hauge S, Lundteigen MA. Guideline for follow-up of Safety Instrumented Systems (SIS) in the operating phase, Ed. 3 (APOS H3). SINTEF; 2023. Report No.: SINTEF Report 2023:00107.

3. Hauge S, Kvam, E, APOS working group. Specification for standardised electronic SRS. 2023.

4. Hauge S, Lundteigen MA, Ottermo MV, Lee S, Petersen S. Information model for functional safety (APOS H5). 2023. Report No.: SINTEF Report 2023:00109.

5. Ottermo MV, Hauge S, Håbrekke S. Reliability data for safety equipment. PDS data handbook. 2021.

6. ISO 14224:2016 Petroleum, petrochemical and natural gas industries — Collection and exchange of reliability and maintenance data for equipment. 2016.

7. IEC 61511:2016, Functional safety - Safety instrumented systems for the process industry sector [Internet]. Available from: https://webstore.iec.ch/publication/24241

8. ISO/TR 12489:2013, Petroleum, petrochemical and natural gas industries — Reliability modelling and calculation of safety systems [Internet]. Available from: https://www.iso.org/standard/51456.html

9. Lane L, Howard C, Hapke HM. Natural Language Processing in Action. Manning publications; 2019.

10. Nadeau D, Sekine S. A Survey of Named Entity Recognition and Classification. 91634. 2007 2007;30(1):3–26.

11. IEC 61508:2010, Functional safety of electrical/electronic/programmable electronic safety-related systems [Internet]. Available from: https://webstore.iec.ch/publication/5515

12. Brundage MP, Sexton T, Hodkiewicz M, Dima A, Lukens S. Technical language processing: Unlocking maintenance knowledge. Manufacturing Letters. 2021 Jan 1;27:42–6.

13. Rausand M. Reliability of Safety-Critical Systems: Theory and Applications [Internet]. Wiley; 2014. Available from: https://books.google.no/books?id=I9rjAgAAQBAJ

14. Maintenance Notification | SAP Help Portal [Internet]. [cited 2022 Sep 28]. Available from: https://help.sap.com/docs/SAP_ERP_SPV/f0356ff01a4748ceb8491bcde7602dc6/d484b8535c39b44ce10000000a174cb4.html?version=6.06.19

15. NORSOK I-002:2021, Industrial automation and control systems [Internet]. 2021. Available from: https://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=1382300

16. Nadkarni PM, Ohno-Machado L, Chapman WW. Natural language processing: an introduction. J Am Med Inform Assoc. 2011;18(5):544–51.

17. Hutchins WJ. Machine Translation: A Brief History. In: Concise History of the Language Sciences [Internet]. Elsevier; 1995 [cited 2022 May 12]. p. 431–45. Available from: https://linkinghub.elsevier.com/retrieve/pii/B9780080425801500660

18. ELIZA—a computer program for the study of natural language communication between man and machine | Communications of the ACM [Internet]. [cited 2022 Oct 21]. Available from: https://dl.acm.org/doi/10.1145/365153.365168

19. Winograd T. Procedures as a Representation for Data in a Computer Program for Understanding Natural Language. 1972.

20. Johri P, Khatri SK, Al-Taani AT, Sabharwal M, Suvanov S, Kumar A. Natural Language Processing: History, Evolution, Application, and Future Work. In: Abraham A, Castillo O, Virmani D, editors. Proceedings of 3rd International Conference on Computing Informatics and Networks. Singapore: Springer; 2021. p. 365–75. (Lecture Notes in Networks and Systems).

21. Chapman WW, Nadkarni PM, Hirschman L, D'Avolio LW, Savova GK, Uzuner O. Overcoming barriers to NLP for clinical text: the role of shared tasks and the need for additional creative solutions. J Am Med Inform Assoc. 2011 Sep;18(5):540–3.

22. Fagan F. Natural Language Processing for Lawyers and Judges. MLR. 2021;(119.6):1399.

23. Dima A, Lukens S, Hodkiewicz M, Sexton T, Brundage MP. Adapting natural language processing for technical text. Applied AI Letters. 2021;2(3):e33.

24. Brundage MP, Sexton T, Hodkiewicz M, Dima A, Lukens S. Technical language processing: Unlocking maintenance knowledge. Manufacturing Letters. 2021;27:42–6.

25. Ottermo MV, Håbrekke S, Hauge S, Bodsberg L. Technical Language Processing for Efficient Classification of Failure Events for Safety Critical Equipment. PHM Society European Conference. 2021 Jun 29;6(1):9–9.

26. Woods C, Selway M, Bikaun T, Stumptner M, Hodkiewicz M. An ontology for maintenance activities and its application to data quality | www.semantic-web-journal.net [Internet]. [cited 2022 Oct 29]. Available from: https://www.semantic-web-journal.net/content/ontology-maintenance-activities-and-its-application-data-quality-0

27. Stewart M, Liu W, Cardell-Oliver R. Redcoat: A Collaborative Annotation Tool for Hierarchical Entity Typing. 2019. 193 p.

28. Snowball [Internet]. [cited 2022 Dec 14]. Available from: https://snowballstem.org/

29. SEARCH, SEARCHB functions [Internet]. [cited 2022 Nov 2]. Available from: https://support.microsoft.com/en-us/office/search-searchb-functions-9ab04538-0e55-4719-a72e-b6f54513b495

30. MATCH function [Internet]. [cited 2022 Nov 2]. Available from: https://support.microsoft.com/en-us/office/match-function-e8dffd45-c762-47d6-bf89-533f4a37673a