Holistic Life-Cycle Machinery models for facilitation Refurbishment & Remanufacturing

September 2020 - M12

AUTHORS: ELIAS MONTINI, NIKO BONOMI, LUDOVICA ROSSI, ANDREA BARNI, MARCO PIROTTA, FABIO DANIELE

DATE: 30.09.2020

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°869884

Technical References

Project Acronym	RECLAIM
Project Title	RE-manufaCturing and Refurbishment LArge Industrial equipMent
Project Coordinator	HARMS & WENDE GMBH & CO KG
Project Duration	01/10/2019 – 31/03/2023

Deliverable No.	2.5
Dissemination level [1]	PU
Work Package	2
Task	2.5
Lead beneficiary	SUPSI
Contributing beneficiary(ies)	SUPSI
Due date of deliverable	30.09.2020
Actual submission date	30.09.2020-

[1] PU = Public
PP = Restricted to other programme participants (including the Commission Services)
RE = Restricted to a group specified by the consortium (including the Commission Services)
CO = Confidential, only for members of the consortium (including the Commission Services)

Document history

V	Date	Beneficiary	Authors
1	21/01/2020	SUPSI	Elias Montini
2	04/06/2020	SUPSI	Elias Montini, Andrea Barni, Marco Pirotta
3	30/07/2020	SUPSI	Elias Montini, Fabio Daniele, Ludovica Rossi
4	30/08/2020	SUPSI	Andrea Barni, Niko Bonomi
5	10/09/2020	SUPSI	Niko Bonomi, Elias Montini, Ludovica Rossi
6	15/09/2020	CERTH	Nikolaos Kolokas
7	15/09/2020	SUPSI	Andrea Barni
8	30/09/2020	SUPSI	Elias Montini

Summary

This document reports the first twelve months' activities of RECLAIM's Task 2.5. As preliminary work, a wide investigation has been made on the management of failures and related metrics in manufacturing companies. Then, a detailed exploration on data-driven approaches, reliability estimation and failure analysis methods have been reported. In this context, also analyzing the results of theT2.1's survey and existent solutions supporting such kind of analysis methods, some limitations of the current solutions aimed at supporting companies, in particular those with less experience and competencies on the failure and maintenance management topic, have been identified. To cover these limits, a procedure and a tool have been proposed, aiming to support companies, in particular SMEs, which have lack of competences and no or weak approaches to maintenance and failure management, in starting their journey toward the digitalization of assets (e.g. machines, tools) health and failure management. The functions that the tool provide support a classic approach to acquisition and elaboration of data to support maintenance management, but are integrated in a software tool enabling also non-expert users to approach it. The main functionalities that it provides are:

- Collecting structured failure data;
- Applying Failure Mode Effect Analysis through a guided procedure;
- Applying Reliability Block Diagram analysis.

The tool is accessible at http://isteps-sps-01.supsi.ch/reclaim/login. Numerous features are already available. More will be integrated in future releases.

Disclaimer

Table of Contents

Acronyms

Acronym	Explanation
AD	Architectural Description
API	Application Programmer Interface
D	Deliverable
T	Task
DSF	Decision Support Framework
IoT	Internet of Things
IT	Information Technology
M	Month
UML	Unified Modeling Language
WP	Work Package
MTTF	Mean Time To Failure
MTBF	Mean Time Between Failure
MTTR	Mean Time To Repair
RBD	Reliability Block Diagram
RAT	Reliability Analysis Tool
FMEA	Failure Mode Effect Analysis
FTA	Failure Tree Analysis

1. Introduction

One of the top priorities of maintenance managers is to ensure maximum operational availability of their equipment, as well as keeping equipment operations safe and efficient. Understanding the calculations and use of failure metrics will enable maintenance professionals to determine, with greater accuracy, when a critical asset is most likely to fail. If the firms want to plan a system based on predictive maintenance, the correct use of these indicators is fundamental.
Developing the analysis of the most adopted methodologies to obtain a reliable estimation of machine average life expectancy is crucial to have a vision about all methodologies concerning life expectancy of equipment, machines and devices. This topic is part of vast sectors of industry research, which include Remaining Useful Life estimation, predictive maintenance, life cycle management, failure measurements and reliability assessment.
This document describes the activities carried out in Task 2.5, aiming at:

- Investigating the current approaches on failure metrics and monitoring applied to improve assets' health and production systems' performance;
- Identifying the main gaps of existent solutions in order to define functionalities and design a specific tool capable to introduce companies that do not specific methods to manage failures;
- Developing a tool aiming to support companies in introducing basic approaches to failure management paving the way to more advanced solutions as those proposed by the RECLAIM project.

This section describes the purpose, the background and the structure of this deliverable, the related terminology, as well as its relationship with other tasks and deliverables.

1.1. Purpose, context and scope

Companies, in particular SMEs, consider maintenance and failures management as a cost and not as a way to improve production system performance. Moreover, despite the industry 4.0 spread out and the hype on advanced approaches such as predictive maintenance, companies still struggle in having basic foundations to support such kind of approaches. To overcome this gap, Task 2.5 (T2.5) investigates the field of failure metrics and management to propose a specific procedure and a tool supporting companies in analyse failure and build the basis to introduce advanced solutions such as those proposed by RECLAIM.

1.2. Background

The vision of RECLAIM is to demonstrate technologies and strategies in support of a new paradigm for the management of large industrial equipment that approaches the end of its design life. This paradigm will substantially reduce the opportunity cost of retain strategies (both money-wise and resource-wise) by allowing relatively-old equipment

that faces the prospect of decommissioning to reclaim its functionalities and role in the overall production system. Such new strategies will contribute to a more sustainable and resource-friendly asset management and, at the same time, offer economic and competitive advantages to the manufacturing sector. To achieve the above, a Decision Support Framework (DSF) will be developed to accumulate knowledge about the health status of machinery and propose innovative methods, tools or services for the appropriate lifetime extension strategy: *1) Refurbishment and Upgrade of industrial equipment* to improve machinery operation and avoid unplanned downtime due to machine failure. *2) Re-manufacturing and Re-use of industrial equipment* to estimate Lifecycle cost and contribute to the re-use of old machinery assets in renewed and new factories. *3) In-situ Repair* to minimise the extra cost and downtime associated with the disassembly and transportation of the machinery. *4) Predictive Maintenance and Fault Diagnosis* to maximise the performance of machinery during its lifetime and provide pragmatic maintenance able to identify equipment failures before they occur.
The RECLAIM Solution with its planned activities addresses currently neglected industrial needs and contributes to unleashing the full potential of sustainable, green and smart factories, by empowering the industry to produce components and assembly systems that meet fast changing requirements. RECLAIM focuses on 100% re-use of equipment through flexible and low-cost systems that support the fast and easy process of refurbishment and re-manufacturing. This perspective will develop self-aware and knowledge-based equipment for the collection and management of operation-related information. All the above-mentioned solutions will be demonstrated in real industrial environments to evaluate the lifecycle of the industrial equipment (machines, production lines, robotic systems, etc.) and to implement the appropriate recovery strategies (refurbishment, re-manufacturing, upgrade, re-use, repair, etc.).
The platform will be demonstrated in 5 different pilots belonging to the five end users (i.e., GORENJE, FLUCHOS, PODIUM, HWH, ZORLUTEKS). In particular, alternative industrial machines will be refurbished and/or re-manufactured. The first pilot scenario includes robots cells and white enamelling line. The second pilot scenario comprises cutting machines and the third pilot scenario contains machines for cutting, drilling and finishing. The fourth pilot scenario comprises friction welding machines and finally the fifth pilot scenario includes bleaching machines.

1.3. Relation to other tasks and deliverables

Task 2.5 has a relation with:

- Task 2.1: T2.5 collaborates in the definition of the survey delivered to end-users supporting the definition of questions related to failure metric and monitoring. Survey results have been used to understand in more detail the context and the current status of maintenance, refurbishment and failure management in European companies.
- Task 3.2: T2.5 aims to contribute to the calculation of the Machine Health-index proposed in Task 3.2. The tool developed in T2.5 includes APIs that allow to communicate the obtained failure metrics to be used to calculate Health-index.

1.4. Content and structure of this deliverable

This deliverable sums up the activities carried out in Task 2.5 in the first 12 months of the RECLAIM project. In this task, a detailed investigation on modelling and estimation methodologies of Residual Useful Life and reliability based on data driven techniques was carried out. This allowed exploring the field of manufacturing system reliability estimation, in particular focusing on Weibull analysis. These activities, together with the results of the survey "End-User and System Requirements" contributed to define the requirements and specifications of a tool to support companies in the analysis of their systems. To this end, the task has involved the development of a tool prototype in using Python, relying on Weibull Analysis, which calculates MTTF/MTBF, dev. std., current reliability, expected time to failure, failure probability during mission. From this prototype, a more advanced web-based solution has been to be developed, providing more sophisticated features.

Section 2 provides an overview of most relevant metric and indicators related to failures, reliability and maintenance in the manufacturing industry. These investigations have been completed with Section 3, where the main methodologies, based on data-driven approaches and adopted to calculate such indicators, have been described. Section 4 reports the needs and requirements identified through a review of the State Of the Art related to failure metrics and maintenance in the industry 4.0 era and a survey carried out in collaboration with task 2.1. One of the main outputs of this section is the need for companies, in particular SMEs, of structured and guided approaches to introduce advanced methods for maintenance and assets health management in their production systems. To this end, sections 5 and 6 describe the tool developed in this task and the procedure to use it to support companies in introducing in their shop-floor simple models that can foster the gradual and incremental adoption of advanced digital solutions for maintenance and assets health management.

2. Reliability and failure metrics in manufacturing

Asset performance metrics like reliability, Overall Equipment Effectiveness (OEE), Mean-Time-Between-Failures (MTBF) and Mean-Time-to-Failure (MTTF) are essential for any organisation with equipment-reliant operations. These metrics allow engineering and maintenance managers to track the reliability of the asset and schedule efficient and less expensive maintenance. Moreover, these indicators, combined also with other parameters (e.g. productivity, efficiency, status, production volumes) and other qualitative considerations, allow managers to estimate Remaining Useful Life (RUL) of an asset and to plan future investment on machines and equipment.
In this section, most relevant metrics and indicators to monitor and control failures in manufacturing are presented and described, aiming at identifying, from this complete and detailed overview, the most suitable ones for the RECLAIM context.

2.1. MTTF, MTTR and MTBF

Even if, in some cases, a failure is classified in various degrees (e.g. partial or total), it simply means that a system, a machine, a device or a component can no longer produce the specific desired results. Even if a machine is still in operation and produces items, it has failed if it does not deliver the expected quantities or if the expected quality is not reached.
Three of the most relevant metrics to measure failures impact in manufacturing companies are:

- Mean Time To Failure (MTTF);
- Mean Time Between Failure (MTBF);
- Mean Time To Repair (MTTR).

MTTF is an essential measure of reliability used for non-repairable systems (e.g. sensors, pressure transmitters). It specifies the expected operating time until the failure of a device, unit or system (IEC 60050 (191), 1990), as showed in Figure 1. Commonly MTTF refers to the lifetime of any product or a device. Its value is calculated as:

$$\mathrm{MTTF} = \frac{\sum \text{Units operation time}}{\text{Number of failed units}}$$

MTTF is a crucial metric used to estimate the lifespan of units that are not repairable. It is also essential to reliability engineers when they need to determine how long a component would last as part of a large piece of equipment.

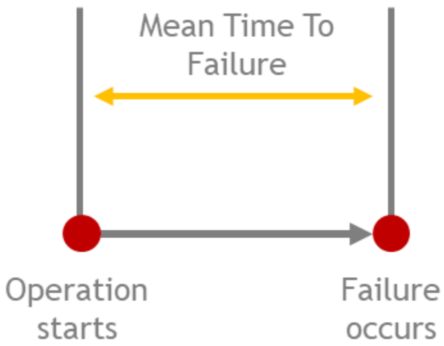


Figure 1 Mean Time To Failure

MTBF specifies the expected operating time between two consecutive failures of a device type in hours (IEC 60050 (191), 1990). The MTBF considers the life cycle of a device that fails repeatedly, then is repaired and returned to service again. MTBF is an essential indicator of expected performance: manufacturers use it as a quantifiable reliability metric and as a critical tool during the design and production stages of many products. Its value is calculated as:

$$\mathrm{MTBF} = \frac{\sum \begin{matrix} \text{Operation time} + \text{Detection Time} + \text{Notification Time} \\ + \text{Diagnosis} + \text{Repair Time} + \text{Testing} + \text{Return to normal conditions Time} \end{matrix}}{\text{Number of failures}}$$

MTBF is usually used for critical assets such as airplanes, safety equipment, and generators, because MTBF is an important indicator of expected performance. Moreover, MTBF can still be applied to calculate the frequency of inspections for preventive maintenance (Christiansen, MTTR, MTBF and MTTF: guide to failure metrics, 2020).

According to ISO 12489 (ISO/TR 12489, 2013), **MTTR** refers not only to the amount of time required to repair a system and restore it to full functionality, but also to the time to detect, notify and diagnose. Typically, every failure event varies in severity. Therefore, while some incidents will require days to repair, others could take mere minutes to fix. Hence, MTTR gives an average of what to expect.

To calculate MTTR, the sum of all the times for detection, notification, diagnosis, repair, testing and return to normal conditions has to be divided by the total number of maintenance actions over a given period:

$$\mathrm{MTTR} = \frac{\sum \begin{matrix} \text{Detection Time} + \text{Notification Time} \\ + \text{Diagnosis} + \text{Repair Time} + \text{Testing} + \text{Return to normal conditions Time} \end{matrix}}{\text{Number of failures}}$$

Every efficient maintenance system always needs to look at how to reduce MTTR as much as possible. That can be done in a few different ways. One approach is through tracking spare parts and inventory levels (thereby saving on downtime while sourcing for parts).

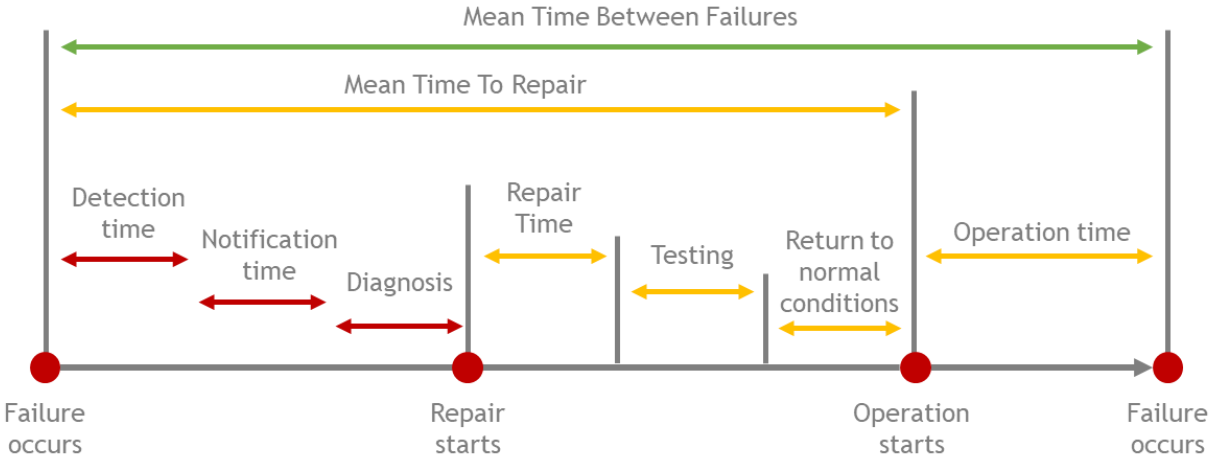


Figure 2 shows the Mean Time Between Failures and the Mean Time to Repair time' components. Depending on the level of details that a company wants to reach, all elements or only few of them can be considered.

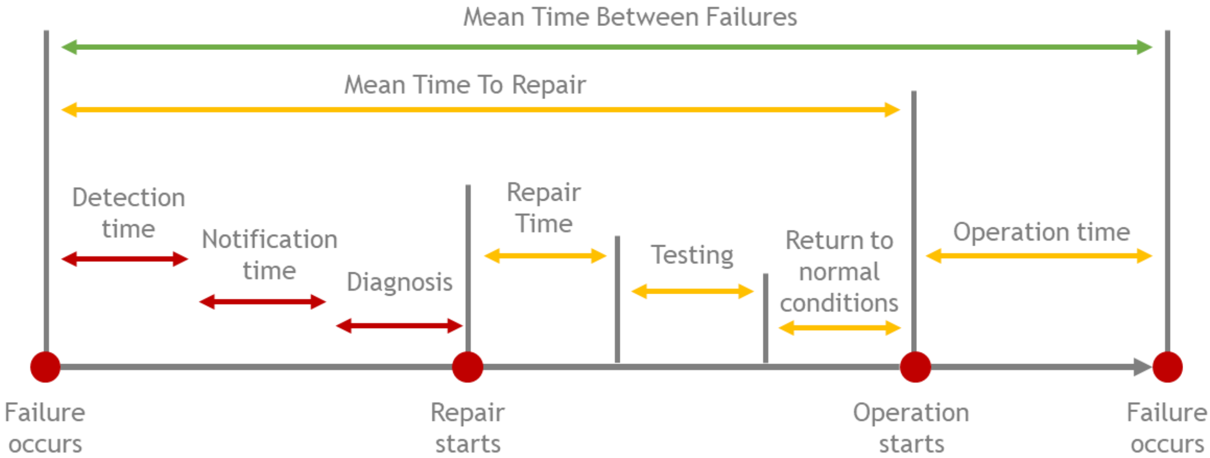


Figure 2 Mean Time Between Failure and Mean Time To Repair

Clearly, all these metrics are statistical and strictly depend on the reliability of the data and on the environment in which these data are collected.

2.2. Reliability

Reliability can be defined as "The probability that a component or system will perform a required function for a given time when used under stated operating conditions. During this correct operation, no repair is required or performed, and the system adequately follows the defined performance specifications." (Coleman, 2020) (Raza, 2020). The reliability function mathematically defines the probability over the operation duration. It is a function of time (or cycles, or miles, or whatever unit of time passing makes sense) (Schenkelberd, 2020).
The reliability function is usually used to answer to "how many units will survive over the warranty period?". If a unit operates 24/7, it will operate 8760 hours in a year. To evaluate the reliability as R(t), it is necessary to calculate R(8760 hours) and it depends

on the particular life distribution characterising the unit. If the result is 0.78 (78%) means that 78 out 100 are expected to survive 8760 hours. It also can mean that one unit has a 78% chance of surviving out to 8760 hours. 22 out of 100 users/customers in the above example are expected to experience unit failure. That may or may not be acceptable. That is a business decision.
The reliability of a component or a system is in general estimated from experience or from test data describing the behaviour given the operating conditions. Statistical methods are usually used for the estimation. In some cases, no prior knowledge of the reliability is available. This will be the case if the unit is designed using a new technology, new materials or subject to new environmental loads. In such cases, it is not possible to use experience and data from components based on other technological design, other materials and other loads.
Reliability could be related to a component or to a system composed by multiple components. In case of system's reliability, it depends on the specifications of individual components, on their configurations, and on redundancy models. The configuration can be series, parallel, or a hybrid of series and parallel connections between system components.
Redundancy models can account for failures of internal system components and therefore change the effective system reliability performance.
It can be observed that the reliability and availability of a series-connected network of components is lower than the specifications of individual components.
For example, two components with 99% availability connect in series to yield 98.01% availability. The converse is true for parallel combination model. If one component has 99% of reliability, then two components combine in parallel yield 99.99% of reliability; three components in parallel yield 99.9999%. Adding redundant components to a system further increases the reliability.
To calculate reliability of a system composed by N components in series, the following formula has to be adopted:

$$R(t) = \prod_{i=1}^{N} R_i\,(t)$$

To calculate reliability of a system composed by N components in parallel, the following formula has to be adopted:

$$R(t) = 1 - \prod_{i=1}^{N} (1 - R_i\,(t))$$

2.3. Remaining useful life (RUL)

The RUL of a system is the time remaining for the system to perform its functional capabilities before irreversible failure. By taking RUL into account, engineers can schedule maintenance, optimise operating efficiency, avoid unplanned downtime, spare parts provision, operational performance, and increase the profitability of the owner of an asset. Xiongzi defined RUL as the duration from the current time to end of useful life for a unit (Xiongzi, Jinsong, Diyin, & Yingxun, 2011). Moreover, RUL estimation is one of the key factors in condition-based maintenance (CBM), and prognostics and health management. RUL estimation plays also an important role in the management of product

reuse and recycle which has strategic impacts on energy consumption, raw material use, pollution and landfill (Mazhar, Kara, & Kaebernick, 2007).
Using a prediction model, a company can estimate time to failure, the risk of failure, and so the system's lifetime. RUL is widely using in both theory and real-life applications by statisticians and engineers.
RUL estimation of a system can be obtained based on its use and performance. This is also known as prognostics (Medjaher, Tobon-Mejia, & Zerhouni, 2012). The RUL of an asset is a random variable: it depends on the current age of the asset, the operating environment and the observed condition monitoring (CM) or health information.

2.4. Other failure metrics

MTTF, MTTR, MTBF, reliability and RUL are the most used failure metrics in the manufacturing industry. However, many others exist and are adopted describing directly or indirectly failures, reliability and maintenance. In particular, different aspects and elements have to be considered and monitored including efficiency, costs and spending, safety and regulation compliancy, assets performance, downtime, work order management, inventory management. In addition to the metrics introduced in the previous sections, others are provided to have a complete overview.

Failure rate
In reliability engineering calculations, failure rate, usually denoted by the Greek letter λ (Lambda), is considered as forecasted failure intensity given that the component is fully operational in its initial condition (Dr. Smith & Simpson, 2016). The formula is given for repairable and non-repairable systems respectively as follows:

$$\text{Failure Rate} = \frac{1}{\text{MTBF}} \text{ or Failure Rate} = \frac{1}{\text{MTTF}}$$

Overall Equipment Effectiveness (OEE)
OEE (Overall Equipment Effectiveness) is mainly used to measure manufacturing productivity. However, the impact that reliability, failures and maintenance have on this metrics are relevant. It identifies the percentage of manufacturing time that a resource is truly productive. An OEE score of 100% means that a machine produces a compliant output, as fast as possible, with no stop time. This means 100% Quality (compliant output).
Measuring OEE is a manufacturing best practice. Indeed, measuring OEE and underlying losses provides important insights on how to systematically improve a manufacturing process. OEE is the single best metric for identifying losses, benchmarking progress, and improving the productivity of manufacturing equipment (i.e., eliminating waste) (leanproduction.com, 2020).

Maintenance Cost per unit Production (MPU)
MPU is a simple ratio which can be measured against benchmarks for each specific industry. A trend upwards in this ratio indicates that there is a problem in the maintenance and reliability performance, while a trend downwards indicates a positive trajectory. A common mistake made to improve this indicator is to cut back maintenance expenses in the short term only to incur major costs later when equipment starts to fail (Kendon, 2019).

$$\text{MPU} = \frac{\text{Cost of maintenance}}{\text{Production achieved}}$$

Maintenance Schedule Compliance (MSC)

MSC can be calculated as a percentage. The target is to have it close to 100%. Relevant deviations in this KPI indicate that there are significant factors preventing the maintenance organization from functioning efficiently.

This could be due to things like breakdowns in equipment or a lack of availability of spares to execute the planned work (Kendon, 2019).

$$\text{MSC} = \frac{\text{Completed maintenance tasks}}{\text{Planned maintenance tasks}}$$

Beside these metrics, others can be also considered (Christiansen, A Ridiculously Simple Guide To Maintenance KPI, 2020):

- **Efficiency**: Maintenance Backlog, Maintenance Overtime, Machine Set-Up Time, Percentage Emergency Work, Overtime Hours, number of Rework Requests;
- **Costs and spending**: Maintenance Cost as Percent of Replacement Asset Value (RAV), Maintenance Cost Per Unit, Utility Consumption (per utility);
- **Safety and regulation compliancy:** Number of Reported Accidents and Incidents, Lost Time Injury Frequency Rate (LTIFR);
- **Assets performance**: Asset Uptime;
- **Downtime**: Production Uptime Percentage, Equipment downtime;
- **Work order management**: Average Time to Complete Work Orders, Percentage of Work Covered by Work Order;
- **Inventory management**: Stock-Out, Inventory Accuracy, Turnover Ratio Slow-moving parts percentage and obsolete parts percentage.

3. RUL and reliability estimation based on data-driven techniques

Data-driven techniques are the most adopted methods for failure metrics estimation due to their ease of implementation and effective results. These techniques are mainly adopted to estimate RUL and do not require detailed knowledge of the analysed system. They rely on the fact that collected data and related features vary with the degradation process of the system, and as a consequence, with its RUL.
Data-driven techniques are useful and reliable when a large amount of data can be transformed into models capable of estimating or support the estimation of failure metrics such as RUL or reliability.
This section aims to introduce advanced data-driven methods including those based on Artificial Intelligent and Machine Learning, but also simpler approaches such as Weibull analysis and reliability analysis. These methods have been evaluated in order to understand which one fits with the RECLAIM context and, in particular, to identify which one fits with the tool that is going to be developed in Task 2.5.

3.1. Data-driven vs Model-based methods

Literature provides several methods to predict the future state of a system (Liao & Köttig, 2014). Categorizing these methods in macro-categories is quite difficult due to the variety of applications. However, the two main categories can be defined as:

- **Data-driven methods**: they are based on the utilization of monitoring data to build behaviour models including the degradation evolution, which are then used to predict the RUL (Dong & He, 2007) (Heng, et al., 2009). This methodology derived from failure data in the historical period so it can be used to predict functioning asset RUL without foreknowledge of the physics of the formation of a component. These techniques do not require particular knowledge of the analysed system and are simpler to apply in the most common manufacturing scenarios compared to model-based approaches.
- **Model-based methods**: they use models generated from fundamental laws of physics to calculate the failure metrics (Chelidze & Cusumano, 2004) (Luo, Pattipati, Qiao, & Chigusa, 2008)

Table 1 highlights the Data-driven and Model-based approaches, specifying also the information necessary for their adoption (Soualhi & Guy, 2019).

	Data-driven	Model-based
System model	Not necessary	Necessary
Historical failures	Necessary	Helpful
Past conditions	Necessary	Necessary
Actual conditions	Necessary	Necessary
Recognition failures methods	Necessary	Necessary
Maintenance history	Not necessary	Helpful

Sensor	Necessary	Necessary

Table 1 Data-driven vs Model-based estimation methods

This comparison highlights the ability of data-driven prediction methods to learn the behaviour of a system using only the features without requiring specific knowledge on the analysed system. However, these methods rely heavily on historical data to determine correlations, establishing models and assessing data trends leading to failure. Data-driven prognostic methods are usually based on the assumption that the monitored features (health indicators of a system e.g. vibrations, temperature) remain relatively unchanged until the occurrence of a failure. In this approach, the monitoring is performed by means of features that indicate the degradation of a system.
Task 2.5 and this document focus on data-driven techniques, given the context and objectives of the RECLAIM project. Industry 4.0 and digitalisation make more and more data available to create digital representations of systems. This data can be used to analyse and apply data-driven techniques to calculate failure metrics.

3.2. Data-driven techniques for RUL estimation

Popular data-driven approaches include deep learning methods such as recurrent neural networks, convolutional neural networks, deep belief networks, and other techniques, such as support vector machines, regression trees, ensemble methods, genetic algorithms, and fuzzy networks for PHM.
The estimation of reliability functions from data collected on the field is very important. Generally, several sets of data are collected, in order to obtain the most details related to occurred failures, such as the number of working cycle or time before the component fails. The goal is to determine the fundamental reliability functions, in particular the failure density function, the survival functions, and the reliability and hazard functions. Data-driven approaches, which rely only on available past observed data and statistical models, can be used universally to achieve the best estimate. The approaches are classified into two broad types of models:

- Models that rely on directly observed state information of the asset;
- Models that do not rely on directly observed state information of the asset.

The RUL of an asset is clearly a random variable and it mainly depends on:

- The current age of the asset;
- The operation environment;
- The observed condition monitoring (CM) or health information.

RUL estimation models provide methods for training a model using historical or sensor data to predict the RUL. These models are useful when historical data and information are available such as:

- Run-to-failure histories of machines similar to the one to diagnose;
- A known threshold value of some condition indicator that points out failure;
- Data about how much time or how much usage it took for similar machines to reach failure (lifetime).

Different types of models exist. The choice on which model to adopt is mainly related to the available data to develop the RUL estimation. Figure 3, elaborated from (www.mathworks.com, 2020), provides an overview of the types of data-driven methods.

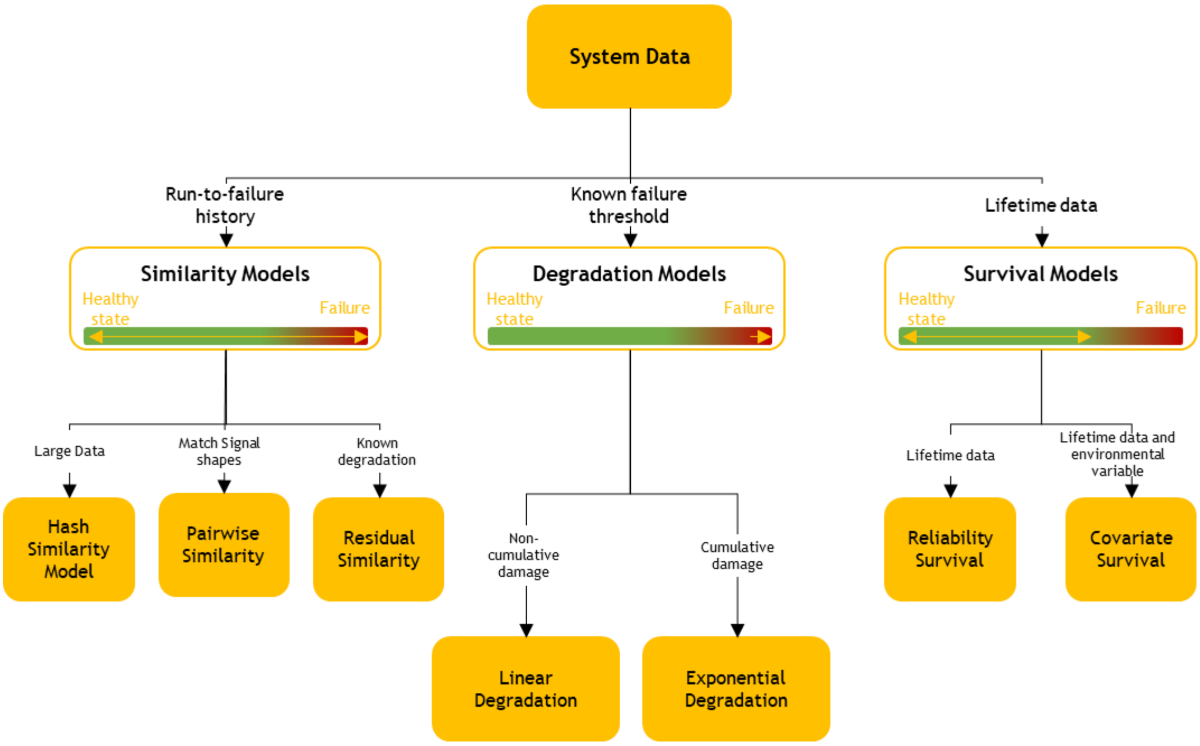


Figure 3 Data-driven model types (edited from mathworks.com)

Table 2 provides details for each data-driven model type.

Models type	Models
Similarity models Similarity models base the RUL prediction of a machine on known behaviour of similar machines from a historical database. Such models compare a trend in test data or condition-indicator values to the same information extracted from other, similar systems. **When to use them?** • If complete history of run-to-failure[1] data of similar	**Hashed-feature similarity model** This model transforms historical degradation data from each member of the ensemble into fixed-size, condensed, information such as the mean, total power, maximum or minimum values, or other quantities. The hashed-feature similarity model is useful when large amounts of degradation data are available. It reduces the amount of data storage necessary for prediction. However, its accuracy depends on the accuracy of the hash function that the model uses.
	Pairwise similarity model Similarity estimation determines RUL by finding the components the historical degradation paths of which are most correlated to that of the test component. In other words, it computes the distance between different time series, where distance is defined as

[1] Run-to-failure data are data that start during healthy operation and end when the machine is in a state close to failure or maintenance.

systems, machines or components are available. • The run-to-failure data shows similar degradation behaviours.	correlation, dynamic time warping, or a custom metric provided. By taking into account the degradation profile as it changes over time, pairwise similarity estimation can give better results than the hash similarity model.
	Residual similarity model Residual-based estimation fits prior data to model such as an ARMA model or a model that is linear or exponential in usage time. It then computes the residuals between data predicted from the ensemble models and the data from the test component. You can view the residual similarity model as a variation on the pairwise similarity model, where the magnitudes of the residuals is the distance metric. The residual similarity approach is useful when your knowledge of the system includes a form for the degradation model.
Degradation models **Degradation models** extrapolate past behaviours to predict the future condition. This type of RUL calculation fits a linear or exponential model to degradation profile of a condition indicator, given the degradation profiles in the ensemble. It then uses the degradation profile of the test component to statistically compute the remaining time until the indicator reaches some prescribed threshold. **When to use them?** There is a known value of condition indicator/s that indicate/s failure (e.g. safety threshold).	**Linear degradation model** It describes the degradation behaviour as a linear stochastic process with an offset term. Linear degradation models are useful when the system does not experience cumulative degradation.
	Exponential degradation model It describes the degradation behaviour as an exponential stochastic process with an offset term. Exponential degradation models are useful when the test component experiences cumulative degradation.
Survival analysis Survival analysis is a statistical method used to model time-to-event data. It uses probability density functions to estimate RUL. **When to use them?**	**Reliability survival model** This model estimates the probability distribution of the failure times. The distribution is used to estimate the RUL of the test component. This model is usually used when the only data available are related to the life span of similar components. For example, you might know how many miles each engine in an ensemble ran before needing maintenance, or how many hours of operation each machine in your ensemble ran before failure.

• There is not a complete run-to-failure data set • Only failures data are available	**Covariate survival model** This model is a proportional hazard survival model which uses the life spans and covariates to compute the survival probability of a test component.

Table 2 Data-driven model types

3.3. Weibull analysis

The Weibull distribution, as a statistical model, is the most common method applied as a lifetime distribution approach to predict RUL. It estimates the life of an average component operating under historically average usage conditions. In reality, the record failure time does not start from zero, therefore it is usually sensible to formulate the lifetime function as a 3-Parameter (μ, λ, θ) Weibull distribution, where μ is the shape, λ is the scale parameter and θ is the location parameter, as defined as follows:

$$\mathrm{f}\,(\mathrm{x}, \mu, \lambda, \theta) = \mu\, \lambda^{-\mu} (x - \theta)^{\mu - 1} \exp\left(-\left(\frac{x - \theta}{\mathrm{MTBF}}\right)^{\mu}\right)$$

Where x is statistical of failure times of all training units. At the same time, the cumulative distribution function (CDF) of that Weibull distribution is presented in the form of:

$$\mathrm{f}\,(\mathrm{x}, \mu, \lambda, \theta) = 1 - \exp\left(-\left(\frac{x - \theta}{\mathrm{MTBF}}\right)^{\mu}\right)$$

To estimate the parameters, the empirical distribution gained from the failure times of the training data is fitted to the equation above using regression method.
The Weibull distribution has a construction from statistics of failure times, provide a degradation model base on Weibull distribution regression and has output in relation to RUL estimation based on the mean residual life of Weibull reliability function (Le Son, 2013).
To better understand the use of the Weibull distribution to estimate the reliability of the system, a literature research has been carried out to understand the main use to. Guo et al. (H. Guo, 2009) presented a three-parameter Weibull failure rate function for wind turbine reliability assessment. The proposed model dealt better with incomplete field failure data than the traditional Weibull model, which in fact is a special case of the introduced three-parameter model. Sutherland et al. (H. Sutherland, 2003) presented an approach for deriving and using accurate Weibull distributions by means of identification of subpopulations of related systems in the maintenance database for condition-based maintenance of motors.
Goode et al. (K. Goode, 2000) presented a RUL prediction method using a statistical model for an application of pumps in a hot strip steel mill. A Weibull distribution was used to model the time to failure. RUL was then calculated during the potential failure to functional failure interval, combining vibration data with reliability data. The reliability functions, which require data associated with individual failure mode, can be developed with sufficient historical data. The method incorporates system age and independent hazards into RUL prediction.

In general, we can say it is easy to build a statistically adequate model; however, the model may not follow the underlying degradation process. The parameters are often selected by experts, and that process is time consuming. The reliability function may require assumptions such as that the times to failure must be independent and identically distributed, all covariates that affect the hazard rate must be included in the model, and the effect of covariates is statistically time dependent.
Weibull analysis was exploited, for example, by APTIV, a global technology company that develops safer, greener and more connected solutions for future mobility, to assess and forecast customer risk due to field warranty failures. In particular, they evaluate the impact of zero-time failures on the life data analysis using Weibull++ software by Reliasoft, comparing different approaches in the handling of those kind of failures data. Another example is constituted by Hill-Rom, a medical technology company that designs, manufactures and distributes medical products and health facilities, including surgical and intensive care beds, lift, etc. Being in the medical industry, risk and liability of new released must be carefully assessed: as a test-to-failure approach, the use of Weibull analysis allows reducing testing times and statistically justifies the design choices.
Another company, operating in the field of waste sorting, exploited instead BQR Digital Field Data Analysis software tool, relying on Weibull analysis, to estimate equipment RUL and to improve the inspections and maintenance plans, based on failure records collected by the Computerized Maintenance Management System (CMMS).

3.4. Reliability and failures analysis

In system reliability engineering, different approaches are exploited to analyse failures and system's reliability. Among these methods, the Fault Tree Analysis and Reliability Block Diagram constitute analytical logical graphical tools to enquire respectively the failure and success space of a system. FMEA constitutes one of the most adopted method to analyse failure modes.

3.4.1 Fault Tree Analysis (FTA)

The Fault Tree Analysis (FTA) is a method to model the pathways within a system that can lead to a foreseeable and undesirable loss event (e.g. system failure) (Vesely & Roberts, 1981). It relies on the Fault Tree Diagram (FTD), which is a top-down structure exploiting Boolean logic to identify the interrelations between a critical system event, called top event, and its causes, called basic events. As represented in Figure 4, FTD consists of two kind of elements, connected through lines:

- **Events**: divided in top undesirable, intermediate events, basic events, external events, undeveloped events, conditioning event.
- **Logic gates** – establishing the relationships between events. The two basic types of gates are:
    - OR gate – it indicates that the output event occurs if at least one of the input events occur. In system reliability terms, OR gate means one component failure is sufficient for the system to fail (it is the equivalent of a series configuration in RBD analysis).
    - AND gate – it indicates that the output event occurs if all input events occur. In system reliability terms, AND gate means all the components

must fail in order for the system to fail (it is the equivalent of a total redundant parallel configuration in RBD analysis)

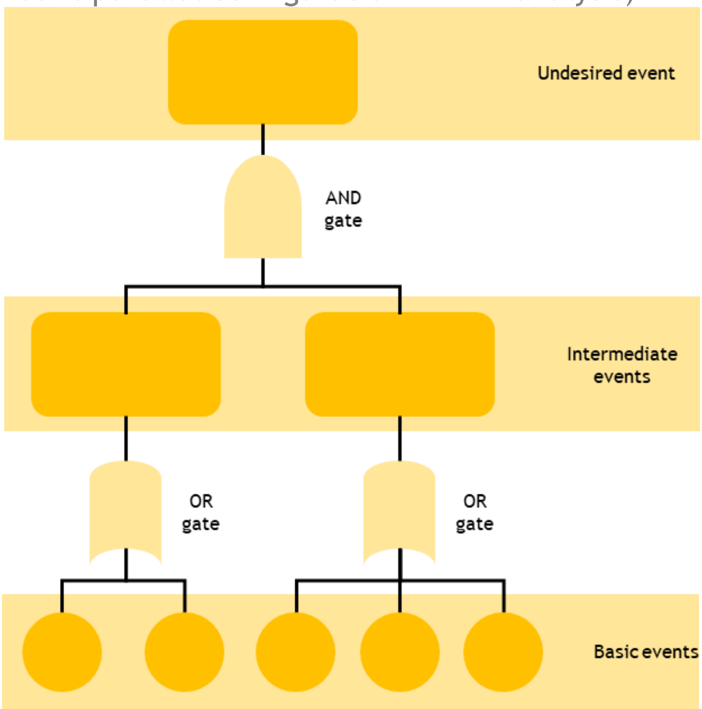


Figure 4 Fault Tree Diagram example

FTA is often carried out in five steps (Lundteigen & Rausand, Fault Tree Analysis (FTA)):

1. **Definition of the problem, system, and boundary conditions of the analysis**: As first, it is necessary to define which aspects of a system are relevant. This selection prescribes what kind of analysis is to be conducted. Then, the definition of the physical external boundaries determines the comprehensiveness of the analysis, delineating system relationships with the environment. On the other hand, the choice of internal boundaries limits the detail of the analysis, establishing the basic interaction within the system. Other aspects to be considered on system boundaries and limits in resolution should be defined accordingly to the purpose of the analysis and to feasibility reasoning, and they may need to be adapted accordingly to any additional information gained during the analysis.
2. **Construction of the FTD:** each tree focuses on one particular undesired event, which constitutes the top event (generally a complete or catastrophic system failure mode). Then, the immediate causes for the occurrence of this top event have to be identified. These are not the basic causes of the event, but intermediate events, which, in turn, have their own immediate causes. Relying on available information and technical data, who applies the FTA has to go down the tree, until reaching, ultimately, the basic events (i.e. the resolution limit of the tree).
3. **Identification of minimal cut sets:** a cut set in a fault tree is a set of basic events the simultaneous occurrence of which ensures that the top event occurs. A minimal cut set is a cut set that cannot be reduced without losing its status as a cut set. The top event occurs if one or more of the minimal cut sets occur. The

main challenge is therefore to identify the minimal cut sets, which can be found writing the set of Boolean equations equivalent to the FTD and substituting through top-down or bottom-up method.

4. **Qualitative analysis of the fault tree:** qualitative analysis of the fault tree includes the identification of a rank of main contributors among the minimal cut sets, according to the idea that failure probabilities associated with minimal cut sets often decrease by order of magnitude as the size of the cut set increases. Component failure probabilities are in general different and depend on testing intervals, downtimes, etc.; therefore, the ranking of minimal cut sets according to size gives only a general indication of importance. Moreover, a check on common causes and dependency of minimal cut set is carried out.
5. **Quantitative analysis of the fault tree:** after the qualitative analysis, a quantitative one can be carried out in case of all minimal cut sets are independent. The probability of the top event to occur is given by:

$$Q_0(t) = 1 - \prod_{j=1}^{k} \left(1 - \check{Q}_j(t) \right)$$

where $\check{Q}_j$(t) is the failure probability of minimal cut set C_j:

$$\check{Q}_j(t) = \prod_{i \in C_j} q_i(t)$$

In practice, the minimal cut sets are dependent since the same basic events may belong to several minimal cut sets: this type of dependency is called "positive dependency". Thus, the formula above overestimates the actual failure probability, but it still can be used as a conservative approximation, as follows:

$$Q_0(t) \leq 1 - \prod_{j=1}^{k} \left(1 - \check{Q}_j(t) \right)$$

Example (Fault Tree Analysis, s.d.): Figure 5 shows a FMD of water pump failure with its basic events and related probability. The example explains how to calculate the failure probability of the pump.

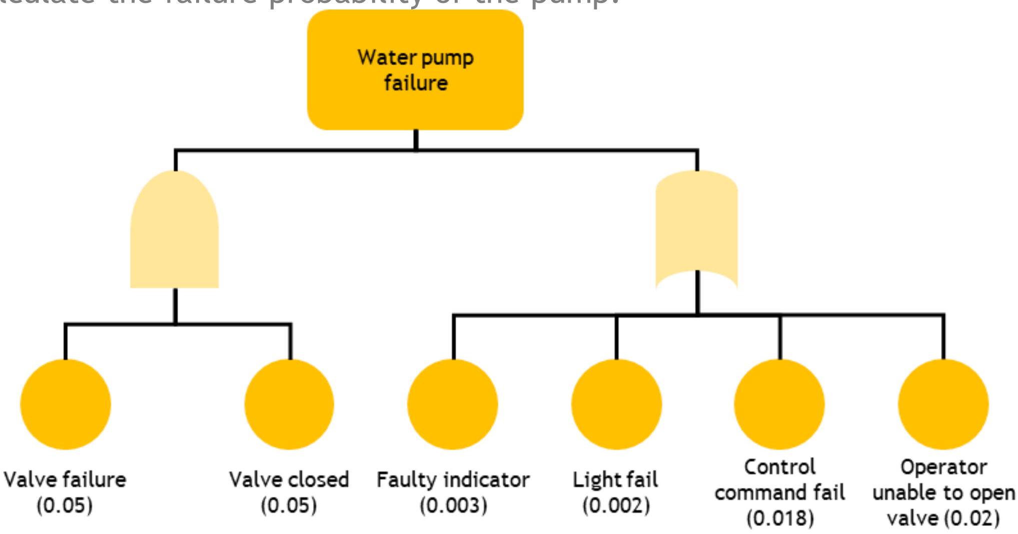


Figure 5 Quantitative analysis FTA: example

The water pump will fail because of value failure and value closed or fault indicator or light fail or control command fail or operator unable to open valve. Since OR gates add and AND gates multiply the probability of pump failure:

$$Q_{pump\ fail} = 1 - (0.05 * 0.05) * (1 - 0.003) * (1 - 0.002) * (1 - 0.018) * (1 - 0.02) = 0.0448$$

Hence, the probability of water pump failure = 4.48%

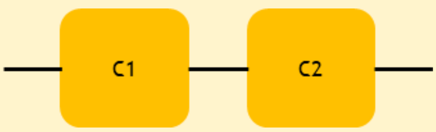
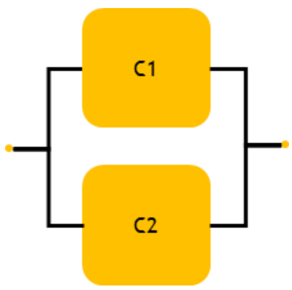
3.4.2 Reliability Block Diagram (RBD)

The Reliability Block Diagram (RBD) is a method to model the relationship among the states of components constituting a system and the success of the system itself in carrying out a specified task (Lundteigen & Rausand, Reliability Block Diagram (RBD)). The RBD shows the logical connection of components, represented by blocks, within a system. The layout represents the reliability structure of the system, but not necessarily its functional one: it can be as a series system, parallel system, or a combined layout. A series configuration represents a system functioning if and only if all the components work, while a parallel configuration represents a system functioning if at least k components out of n, with k ≤ n, work. They are equivalent to, respectively, an OR gate and an AND gate in FTA analysis.

The reliability of a series system is always lower than the one of the weakest components. On the contrary, redundancy allows parallel system reliability to be greater than maximum reliability of one component. Thus, a parallel system can be further classified based on its type and degree of redundancy:

- **Active redundancy system**: all components operate simultaneously:
  - Total redundancy: the system works when at least one component works (k=1);
  - Partial redundancy: k>1 over n components are required for the system to work.
- **Standby redundancy system**: some or all components do not operate continuously but are activated only upon failure of the primary component performing the task.

All the configurations are schematically shown in Table 3.

RDB model	RBD	Semantics
RBD series		Components C1 and C2 are connected in series
RBD parallel (total redundancy)		Components C1 and C2 are connected in parallel, in total redundancy

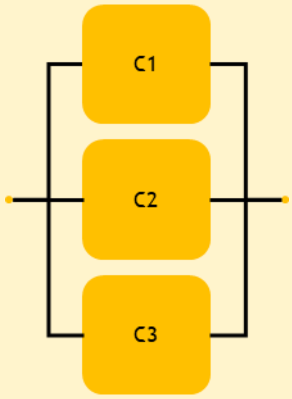
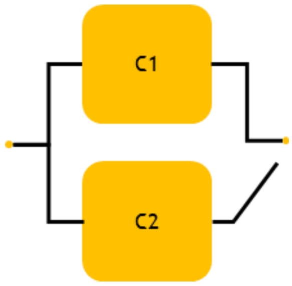
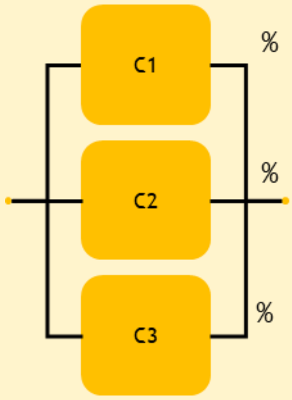
RBD parallel (partial redundancy)		Components C1 … Cn are connected in parallel, in partial redundancy (k over n components are required for system to work)
RBD parallel (stand-by)		Components C1 and C2 are connected in parallel, in total redundancy, with component C2 hold in standby.
RBD parallel multi state (fractioning)		Components C1, C2 and C3 are connected in parallel and they have different capacity, therefore a component's failure involves a loss of capacity corresponding to the impact factor % of the failed component

Table 3 RBD configurations

With respect to fault tree analysis, the RBD method has a success-oriented logic, and it can be used to derive system complexity. Analytical formulas for reliability in the different configuration are showed in Table 4.

Reliability of n series components

Formula	Example
$\boldsymbol{R_s = \prod_{i=1}^{n} R_i}$	$R_1 = 0.99; R_2 = 0.97; R_3 = 0.76$ $R_s = R_1 * R_2 * R_3 \ = \ 0.729$

Reliability of n parallel components - total redundancy (k = 1)

Formula	Example
$\boldsymbol{R_s = 1 - \prod_{i=1}^{n} (1 - R_i)}$	$R_1 = 0.99; R_2 = 0.97; R_3 = 0.76$ $R_s = 1 - (1 - R_1)(1 - R_2)(1 - R_3) \ = \ 0.999$

Reliability of n parallel components - partial redundancy (k ≠ 1)

Formula	Example
General k, n with $R_i = R \ \forall \ i$ A. **k ≥ (n+1)/2:** $$\boldsymbol{R_s = \sum_{i=0}^{n-k} C_{i,n} R^{(n-i)} (1-R)^i}$$ with $\boldsymbol{C_{i,n} = \frac{n!}{(n-1)!i!}}$ B. **k < (n+1)/2:** $$\boldsymbol{R_s = 1 - \sum_{i=n-k+1}^{n} C_{i,n} R^{(n-i)} (1-R)^i}$$	k = 2, n = 3, R = 0.90 $$R_s = R^3 + 3R^2(1-R)$$ $$R_s = 0.972$$
General k, n with $R_i \neq R$ A. inclusion-exclusion principle B. pivoting or factoring technique (recursive algorithm)	k = 2, n = 3, $R_1 = 0.99; R_2 = 0.97; R_3 = 0.76$ $$R_s = R_1R_2R_3 + R_1R_2(1-R_3) + R_1(1-R_2)R_3 + (1-R_1)R_2R_3$$ $$R\_s = 0.990$$

Reliability of n parallel components - standby redundancy

Formula	Example
Hypothesis #1Invalid source specified.: • Block active times to failure are negative exponentially distributed; • Passive failure rates & switching failure rates are assumed to be zero. A. **Equal components, with general k, n** $$\boldsymbol{R_s = e^{-k\lambda t} \sum_{i=0}^{n-k} \frac{(k\lambda t)^i}{i!}}$$ B. **Unequal components:** a) k = 1, n = 2	A. k = 1, n = 3, MTBF = 20.000 h, Mission Time t = 100.000 h $$R(t = 100.000) = e^{-\frac{1}{20.000}*100.000} \sum_{i=0}^{3-1} \frac{\left(\frac{1}{20.000}*100.000\right)^i}{i!} = 0{,}1247$$ B. k = 1, n = 2, λ_1 = 0,001 failure/h, λ_2 = 0,002 failures/h, Mission Time t = 200 h $$R(t = 200) = \frac{0{,}002 * e^{-0{,}001*200}}{0{,}002 - 0{,}001} + \frac{0{,}001 * e^{-0{,}002*200}}{0{,}001 - 0{,}002} = 0{,}967142$$

$$R_s(t) = \frac{\lambda_2 e^{-\lambda_1 t}}{\lambda_2 - \lambda_1} + \frac{\lambda_1 e^{-\lambda_2 t}}{\lambda_1 - \lambda_2}$$

b) k = 1, n = 3

$$R_s = \frac{\lambda_2 \lambda_3 e^{-\lambda_1 t}}{(\lambda_2 - \lambda_1)(\lambda_3 + \lambda_1)} + \frac{\lambda_1 \lambda_3 e^{-\lambda_2 t}}{(\lambda_1 - \lambda_2)(\lambda_3 - \lambda_2)} + \frac{\lambda_1 \lambda_2 e^{-\lambda_3 t}}{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)}$$

Hypothesis #2:

- Failure rates are not constant in time (failure distribution is not exponential);
- Passive failure rates & switching failure rates are assumed to be zero;
- Non-standby component of the system has CDF F(t) and (n-1) identical backup components operate in sequence until the last one fails

Total system lifetime is the sum of n identically distributed random lifetimes, each having CDF $F_i(t)$: $F_s(t)$ can be evaluated through convolution formulas. Invalid source specified.

$$F_2(t) = \int_0^t F(u) f(t-u) du$$

$$\dots$$

$$F_i(t) = \int_0^t F_{i-1}(u) f(t-u) du$$

$$\dots$$

$$F_n(t) = \int_0^t F_{n-1}(u) f(t-u) du = F_s(t)$$ where f(t) is the PDF of F'(t). $$R_s(t) = 1 - F_s(t)$$	

Table 4 RBD analytical formulas

3.4.3 Failure Mode and Effects Analysis (FMEA)

Failure Mode and Effects Analysis (FMEA) is a systematic approach to assess potential system failures and the resulting consequences of those failures. The main objective of a FMEA is to evaluate the risk associated with the identified failure effects and to identify actions to detect, prevent, or mitigate those deemed most critical (Stamatis, Failure mode and effect analysis: FMEA from theory to execution, 2003).

Thus, every FMEA starts from the identification of potential failure modes and related causes and effects. Then, an assumption is made: different problems have different priorities according to the risk associated to them. Three components help to define the priority of failures:

- Occurrence (O): frequency of the failure.
- Severity (S): seriousness (effects) of the failure.
- Detection (D): ability to detect the failure before it reaches the customer.

The usual way to define the value of these components is to use numerical scales, called risk criteria guidelines, which can be qualitative and/or quantitative.

The prioritization of failure modes is then based on the risk priority number (RPN), which is the product of the three above components:

$$RPN = O * S * D$$

Finally, FMEA provides for problem follow-up and corrective action.

There are different types of FMEA, according to which phase and system-level is under analysis. The three most common types are (Carlson, 2012):

- **System FMEA**: it is used to analyse the early concept and design stage of a system, with the aim of improving it. System FMEA highlights the potential failure modes between the function of a system caused by system-related deficiencies, focusing on:
    - System safety and system integration issues
    - Interfaces between subsystems or with other systems
    - Interactions between subsystems or with the surrounding environment
    - Single-point failures, namely where a single component failure can result in complete failure of the entire system
    - functions and relationships that are unique to the system as a whole, i.e., do not exist at lower levels, and could cause the overall system not to work as intended

  The outputs of the system FMEA are:

    - a potential list of failure modes ranked by the RPN
    - a potential list of system functions that could detect potential failure modes
    - a potential list of design actions to eliminate failure modes, safety issues, and reduce their occurrence

  Thus, system FMEA helps in choosing the optimum design alternative, determining system redundancy and defining the basis for system-level diagnostic procedures.
- **Design FMEA:** it is used to analyse the designed product before the release to manufacturing, with the aim of improving the design of subsystems or components. Assuming that the product will be manufactured according to specifications, the design FMEA highlights potential failure modes caused by design-related deficiencies, focusing on:
    - ensuring product operation is safe and reliable during the useful life of the equipment
    - analysing interfaces between adjacent components.

  The outputs of the design FMEA are:
    - A potential list of failure modes ranked by the RPN
    - A potential list of critical and/or significant characteristics and related recommended actions
    - A potential list of design actions to eliminate failure modes, safety issues, and reduce their occurrence
    - A potential list of parameters for appropriate testing, inspection, and/or detection methods.

  Thus, design FMEA assists in the evaluation of components-level design requirements and alternatives, establishing improvement actions and documenting the rationale of changes. The analysis provides further information to help through the components design verification and testing phases.
- **Process FMEA**: it is used to analyse the manufacturing and/or the assembly process, with the aim of improving the design of manufacturing and/or assembly process itself. Assuming the design is sound, the process FMEA highlights potential failure modes caused by process-related deficiencies, focusing on manufacturing and assembly operations, shipping, incoming parts, transporting of materials, storage, conveyors, tool maintenance, and labelling, and ensuring the product is built to design requirements in a safe manner, with minimal downtime, scrap and rework. The outputs of the process FMEA are:
    - A potential list of failure modes ranked by the RPN
    - A potential list of critical and/or significant characteristics
    - A potential list of recommended actions to address the critical and significant characteristics

Thus, process FMEA offers a corrective action plan, documenting the rationale of the changes, and helps developing control plans for the manufacturing and/or assembly phases.

The results of System FMEA are the input for Design FMEA, the output of which in turn becomes the input for Process FMEA, as can be seen in Figure 6, modified from (Stamatis, Failure Mode and Effect Analysis: FMEA from Theory to Execution, 2003).

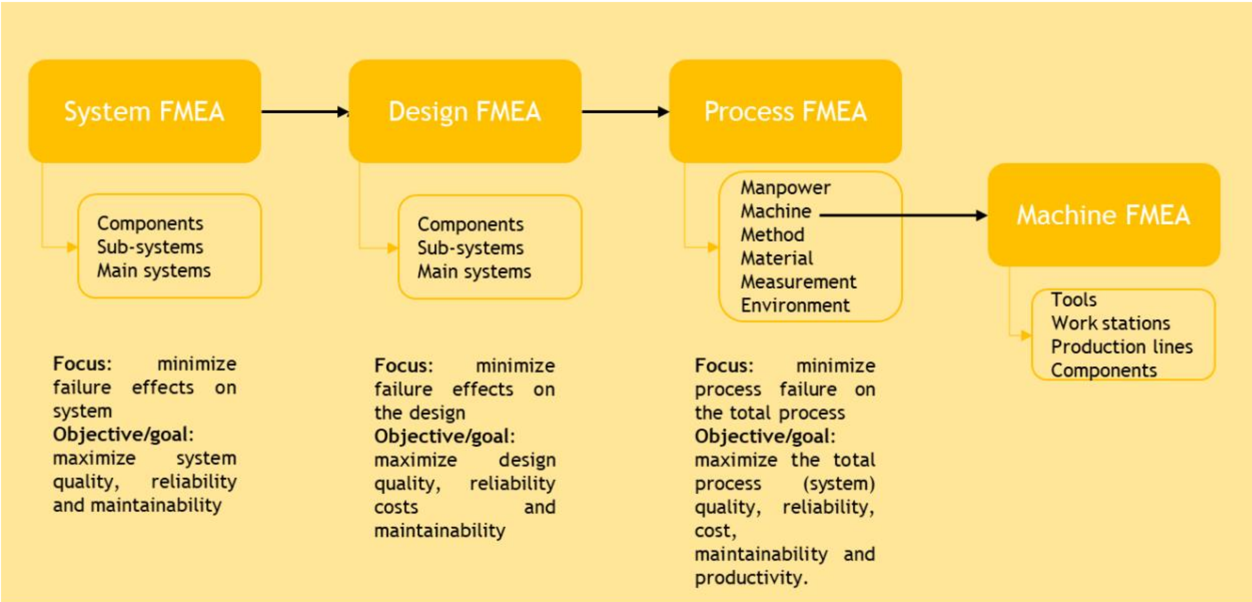


Figure 6 Classification of FMEA typologies

4. Needs and requirements

This section aims to provide an overview of the main reasons that have motivated the design and development of the tool proposed in section 6.

4.1. The impact of industry 4.0 and digitalisation on manufacturing maintenance and failures

Effective maintenance of machines, equipment and tools is a critical factor in delivering quality outputs at a minimal cost. However, this does not come easy. Organizations need to set indicators and quality benchmarks to measure the current effectiveness, predict future performance and use the data collected to understand current status and identify where to make improvements. One way to do this is by using maintenance metrics: these metrics are very important as they can mean the difference between achieving the overall business goals and explaining how unexpected breakdowns or failures caused yet another production delay.

In any asset-intensive industry, including manufacturing, reactive maintenance can wreak havoc on operational efficiency, profit margins, and sustainable competitiveness. Conversely, preventative maintenance can yield significant financial and organizational benefits. In fact, studies have shown that businesses across all sectors of industry can reduce equipment maintenance costs by 12 to 18% by investing in preventative maintenance and tracking failure rates (Dynaway, 2019). A relevant part to implement successful preventative maintenance is the understanding of failure data and the effective use of KPIs.

Manufacturing plants usually integrate various machines and equipment, which have different reliability requirements, failure rates and related effects on production (Das, Lashkari, & Sengupta, 2007). A dedicated maintenance strategy, tailored on the type, health status and reliability requirements of these assets, allows optimizing costs and operations management, achieving minimum breakdown at minimum costs (Bevilacqua & Braglia, 2000) (Wang, Chu, & Wu, 2007). The importance of selecting proper maintenance strategies has been acknowledged in various areas since the second half of the 80s (Arunraj & Maiti, 2010). The most well-known methodologies for this purpose are Reliability Centered Maintenance (RCM), Risk-Based Inspection (RBI) and Risk-Based Maintenance (RBM). In particular, RCM aims to minimize the maintenance cost by balancing the use of different maintenance strategies (Nowlan & Heap, 1978) (Moubray, 2001).

The rising adoption of digital technologies, with sensing and data elaboration infrastructures becoming more affordable, makes advanced maintenance strategies (e.g. predictive and prescriptive maintenance) easier to implement. Though, predictive maintenance cannot be adopted by all the companies: technologies provided by the industry 4.0 framework, and digitalisation in general, reduce entry barriers and implementation costs, but other relevant issues can influence decision-making, leading to a reduced conversion rate in the adoption of the most effective approaches to maintenance:

- Technologies are not enough to realize effective maintenance management. Data needs to be collected, structured and used properly to measure machines' Overall Equipment Effectiveness (OEE), failure occurrences, maintenance activities, etc. Companies still lack effective systems capable to support structured and accurate data collection (Shalabi & Turkan, 2017) (Confalonieri, Barni, Valente, Cinus, & Pedrazzoli, 2015).
- Maintenance is often considered a cost centre and not a profit-generating function. Problems in production processes due to failures and maintenance reduce productivity, increase product cost, even may lead to loss of timely services to customers, and thereby reduce profitability. Accountants think of maintenance in terms of costs, production managers in terms of performance and equipment availability. There is a lack of common language, which does not enable the different functions in the organisation to understand each other (Alsyouf, 2007).
- Many companies, in particular SMEs, have a lack of knowledge on the requirements necessary to adopt advanced maintenance strategy (Baglee, Gorostegui, Jantunen, Sharma, & Campos, 2017).
- Managers are not aware of how to do maintenance better, or if they are, they think that the necessary investment is too high (Baglee, Gorostegui, Jantunen, Sharma, & Campos, 2017).
- Support systems such as Computer Maintenance Management Systems (CMMS) are being used with partial success in many industries due to their limited capabilities. They generate work orders and job sheets, assign maintenance personnel to their tasks, manage inventory to a certain extent and produce maintenance schedules based on a set of rules. However, these systems do not have the capability of diagnosis or prognosis to define or support the definition of the rules on which maintenance management relies on (Labib, 2004).
- Tools and methods supporting maintenance strategy selection, on one hand, need relevant effort to be exploited, requiring a huge amount of information, time to be understood and applied; on the other hand, more practical and qualitative approaches do not allow to make accurate selection (Tahir, Prabuwono, & Aboobaider, 2008).

4.2. End-users needs and requirements analysis

In the activities carried out in Task 2.1, a survey has been developed and delivered to more than 50 companies in order to collect end users' needs and requirements related to maintenance and refurbishment of machines and equipment. Lessons learnt from this survey are reported in section 4.4. This survey is composed by 43 questions. Among these, 4 questions & answers have been proposed by Task 2.5, in order to depict the current scenario on asset management and monitoring.

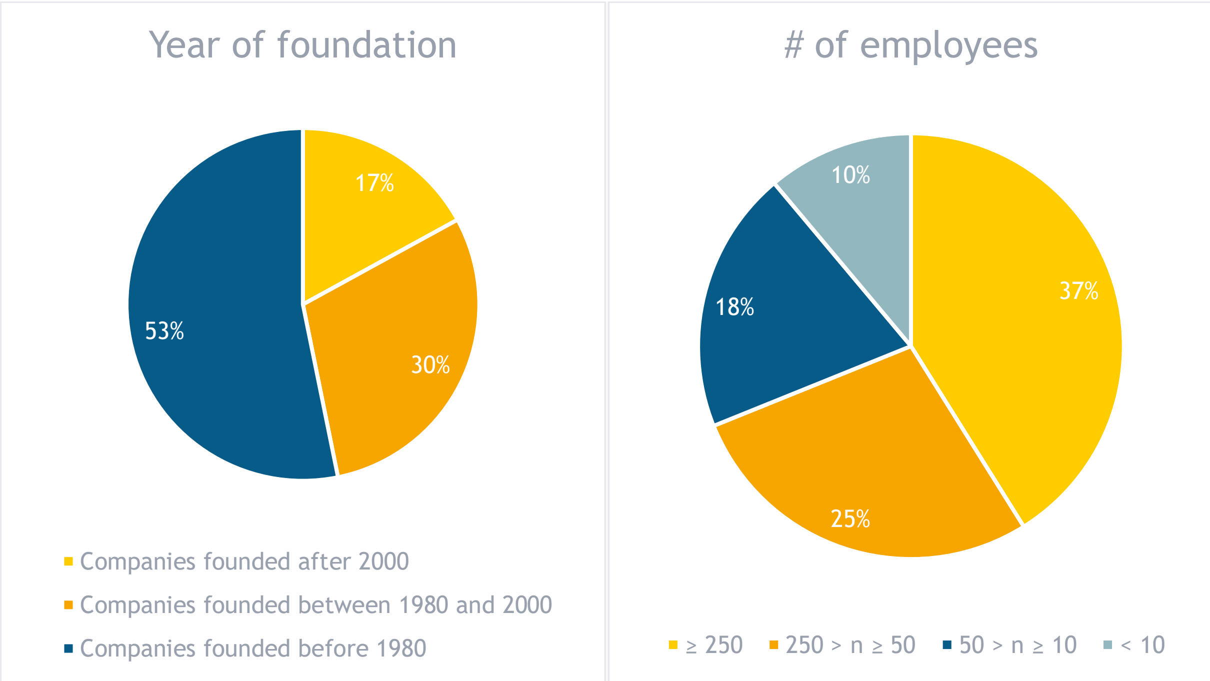


Figure 7 provides an overview of the age and size of the companies which have answered to the survey. The 83% has more than 20 years. The 37% have more than 250 employees.

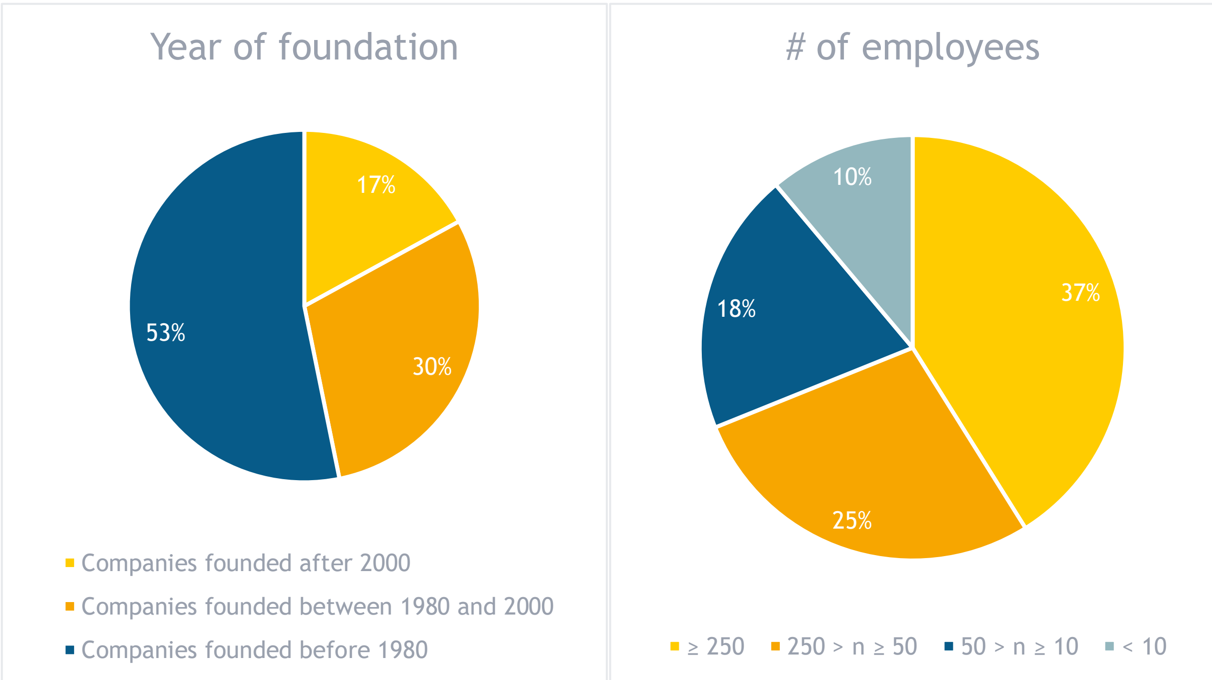


Figure 7 Interviewed companies year of foundation and number of employees

Figure 8 describes the type of machines owned by interviewed companies. These own mainly machining, forming and moulding machines. The 31% of these machines has more than 20 years and only the 14% has less than 5 years (7% less than 1 year, 7% from 1 to 5 years).

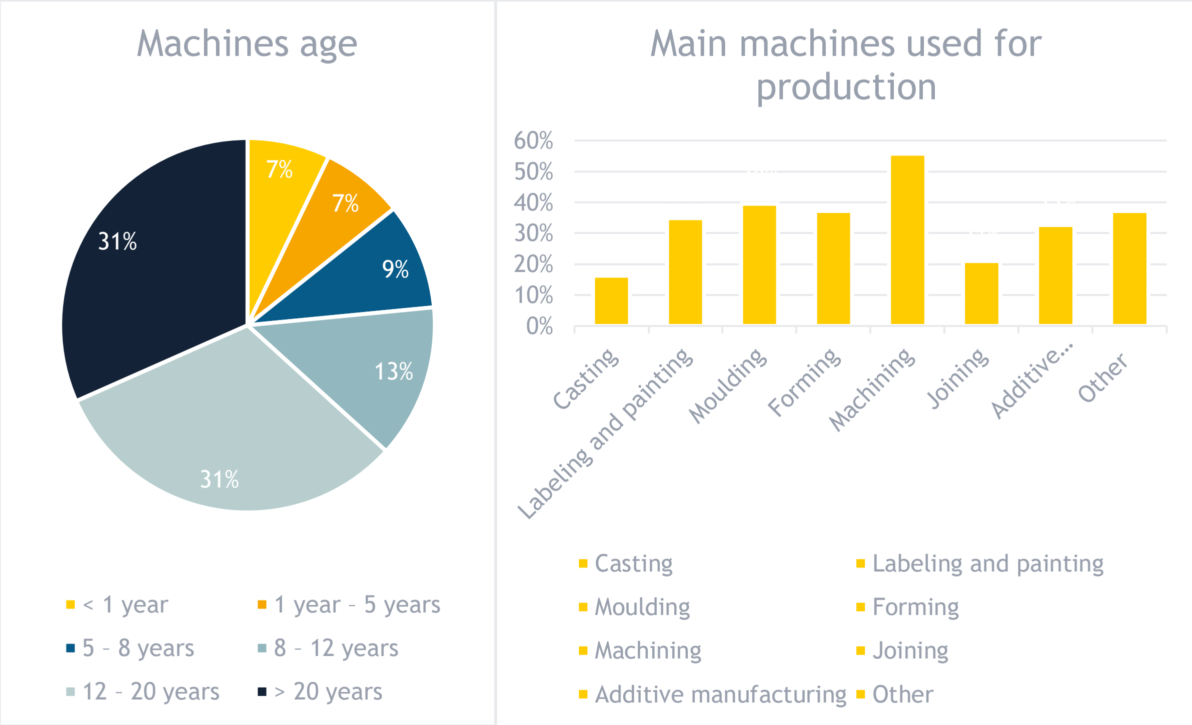


Figure 8 Main machines age and types

Each question of the interview is presented according to the following structure:

- Question: it details the question asked to the interviewed;
- Answer: it details the answer provided by the interviewed, mainly with graphs and tables.

4.2.1 Survey results

Question: *How reliable do your assets are and need to be (average OEE level)?*
Answer: Interviewed companies reported different levels of current and target OEE levels. These are resumed in Figure 9. Unknown means that the companies currently do not measure it. NA means that the interviewed person does not know the meaning of OEE. As it is possible to see, nearly the 50% of the interviewed does not give importance to the evaluation of OEE metric, with a relevant percentage (29%) that does not measure OEE and the 19% does not even know OEE meaning. Moreover, among the ones that measure their productivity through this indicator, nearly the 50% has an OEE lower than 90%.

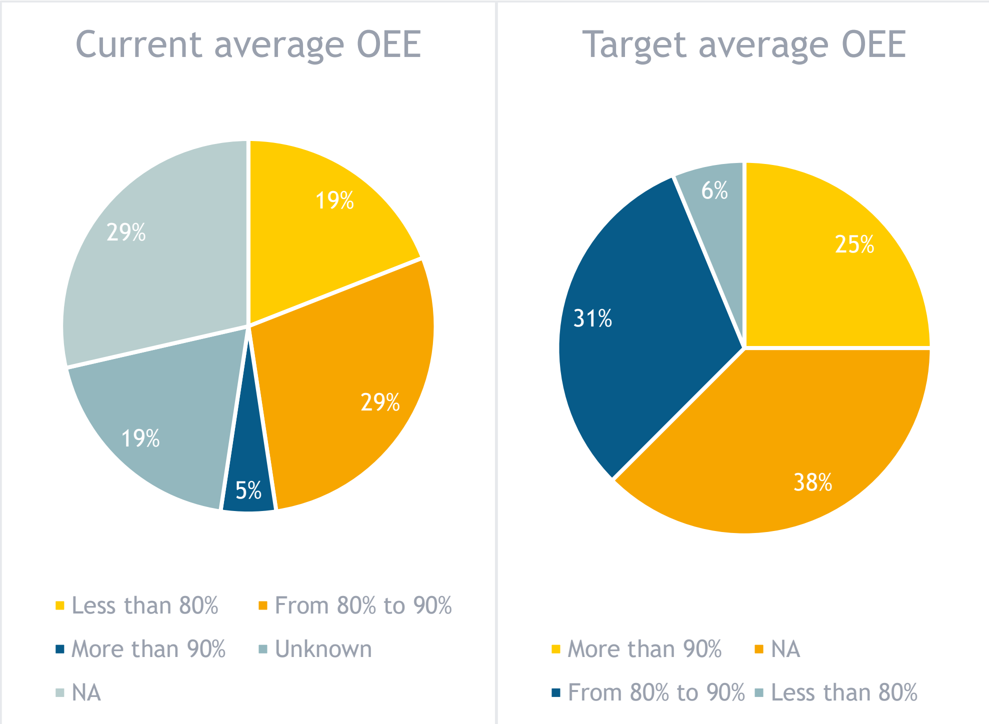


Figure 9 Respondents' asset current and target average OEE

Question: *Which of the following approaches does mostly fit with the current maintenance approach of your company? Please rank the 5 proposed strategies, assigning 5 points to the one that fits most, 4 to the second, and so on.*
Answer: The average values between all answers have been represented in Figure 10. As it is possible to see, prescriptive and predictive maintenance strategies, despite their potentiality, have the lowest scores of adoption. On the contrary, reactive and corrective strategies, which are usually the less valuable approaches, have the highest scores, meaning that for the majority of SMEs the adoption of advanced maintenance strategies there is still a long way to go.

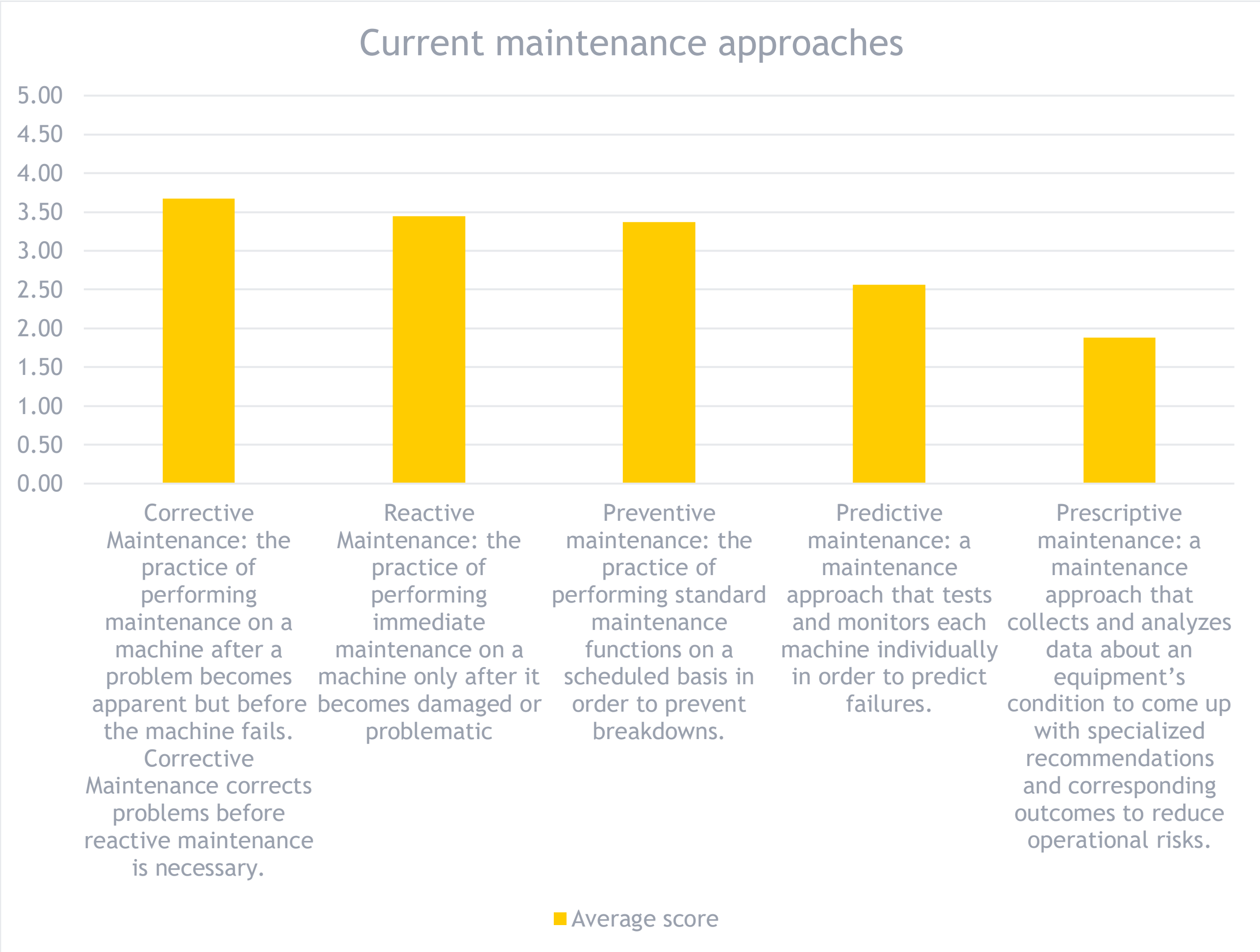


Figure 10 Currently adopted maintenance approaches

Question: *Which are the most relevant barriers you faced trying to achieve an advanced asset management approach? Please add any relevant detail.*

Answer: Figure 11 shows the most relevant barriers the interviewed companies experience in the attempt of transforming their current asset management approach into an advanced one. Nearly the 60% claims to have no knowledge of previous successful cases of companies gaining economic margin through the adoption of these new strategies. Lack of guarantees in this regard has slowed down the investments and, consequently, the transformation. Second barrier observed, in terms of percentage, is related to the difficulty that is intrinsic in every change of mind-set: times and money are required to reorganize the different involved units and to coordinate their activities. Less concern is related to need of expertise, even if companies complain about the lack of qualified personnel. Being data science a world in continuous evolution, companies are scared of lack of security and related intellectual property thefts coming through the sharing of data.

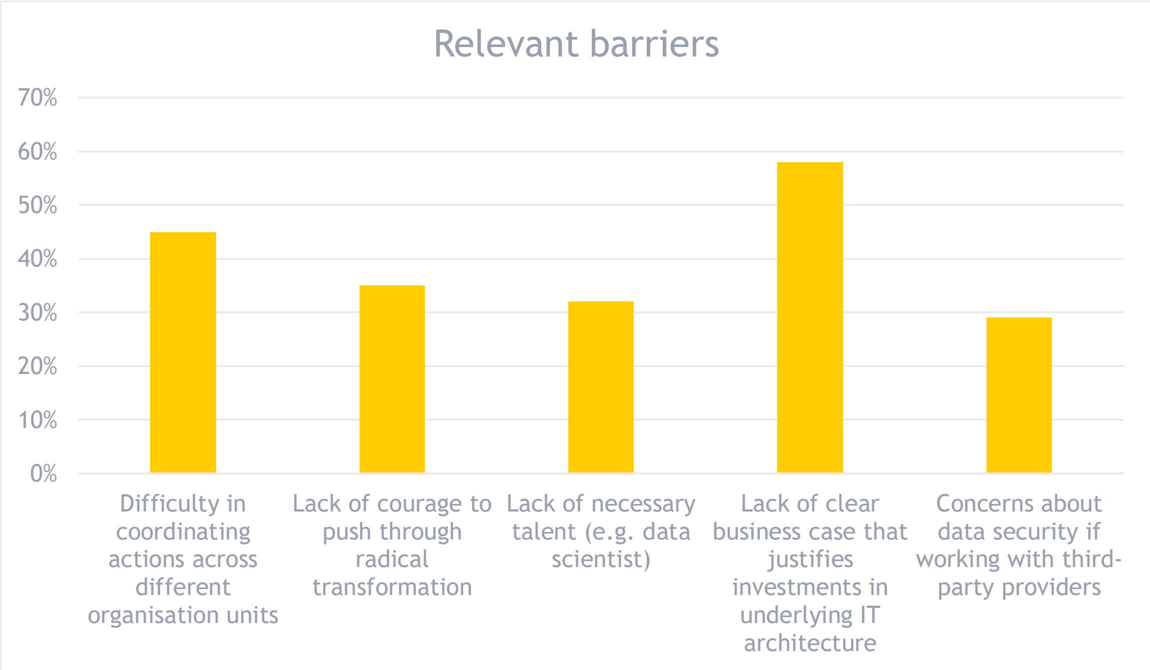


Figure 11 Relevant barriers

Question: *Which of the following activities do you expect to integrate into your daily operations by the next 5 years? Please add any relevant detail.*
Answer: Figure 12 shows the percentages of respondents that expect to integrate in the next 5 years each of the listed activities. Most of the respondents have selected "Analyse and classify assets and their components". However, all the other activities have received more than the 40% of selection. Despite all the uncertainties emerged from previous question&answer, the interviewed companies seem aware of the changes the everyday their work will face in the next future in order to keep competitive on the market.

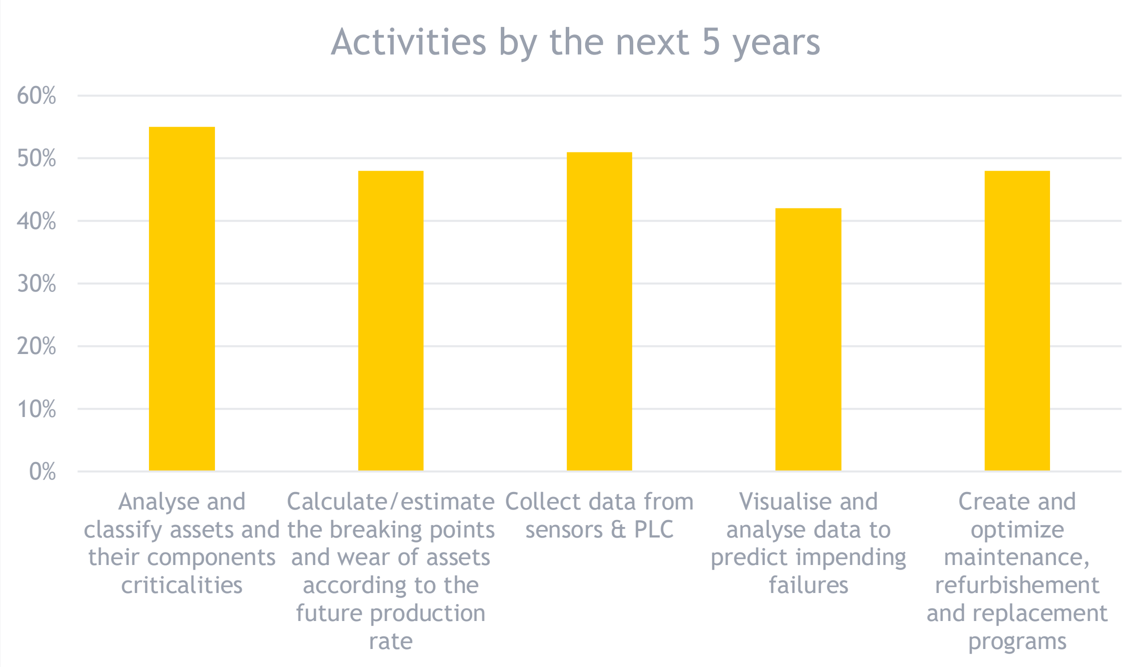


Figure 12 Activities by the next 5 years

4.3. Tools and software supporting the evaluation of assets and machines status

This section provides an overview on existing tools and software supporting reliability analysis, with the aim of highlighting their features to then introduce the innovative aspects proposed by RECLAIM tool in section 4.4.

4.3.1 Reliasoft

ReliaSoft is a software developed by HBM Prenscia, which provides a range of solutions to facilitate a comprehensive set of reliability engineering modelling and analysis techniques (HBM Prenscia, 2020).
In particular, ReliaSoft BlockSim offers a platform for the analysis of systems' reliability, availability and maintainability allowing modelling complex systems and processes using RBDs, FTA or Markov diagrams. BlockSim supports an extensive array of RBD configurations, including load sharing, standby redundancy, phases and duty cycles; RBDs can be built by easy drag-and-drop techniques. Using exact computations and/or discrete event simulation, BlockSim facilitates different analyses for both repairable and non-repairable systems. Its main functionalities are:

- Reliability analysis;
- Identification of critical components (Reliability Importance Measures)
- Optimum reliability allocation;
- Maintainability analysis: determination of optimum preventive Maintenance Intervals, Spare Parts Provisions, etc.;
- System availability analysis: calculation of Uptime, Downtime, Availability, etc.;

- Throughput calculation: identification of Bottlenecks, Estimate Production Capacity, etc.;
- Life Cycle Cost Estimation.

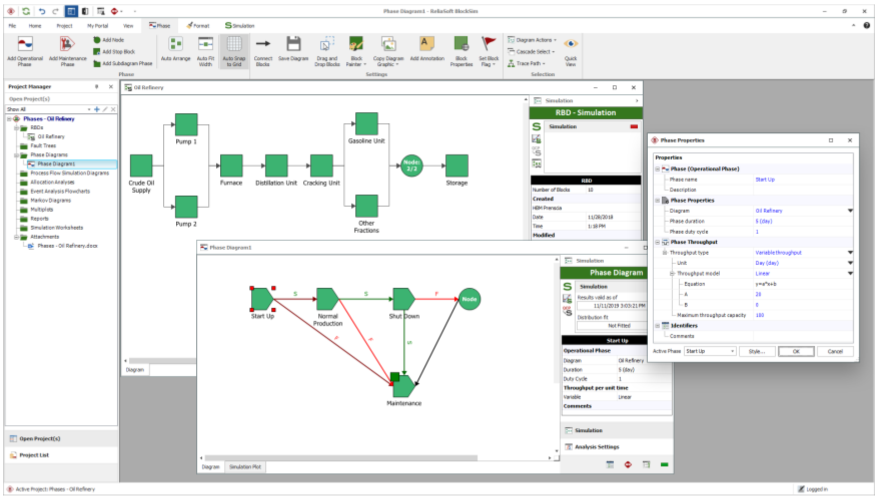


Figure 13 Reliasoft BlockSim

ReliaSoft XFMEA, instead, allows performing different types of FMEA analysis, including Design FMEA, System FMEA, Process FMEA and FMECA with reporting capabilities and risk discovery tool. FMEA analysis can be easily linked to RDBs and FTAs, and transform the FMEA findings into a representative reliability model of the system. The software enables to build a continuous knowledge repository of the FMEA results to be reused throughout the reliability program. XFMEA supports the major industry standards for all types of FMEA analysis; besides providing predefined profiles for the major reporting standards, it is also allowed a degree of interface customization.

4.3.2 Relyence

Relyence Corporation offers a range of software products thought for a variety of sectors, from aerospace to healthcare. Relyence's reliability suite provides a complete toolset targeting reliability and quality objectives (Relyence Corporation, 2020).
Relyence RBD allows to create reliability block diagrams inserting blocks, and arranging them in series, parallel, hot and cold standby configurations.

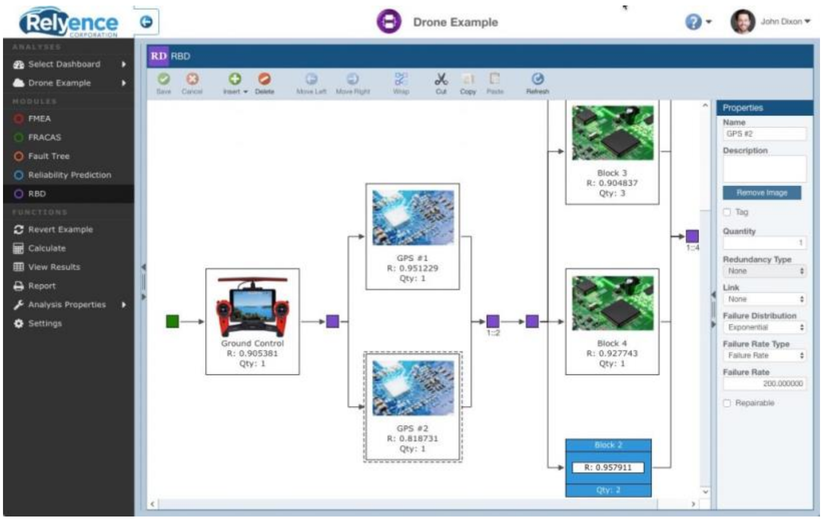


Figure 14 Relyence RBD

By clicking on the blocks, as shown in Figure 15, it is possible to define redundancy type, failure and repair distributions, if applicable, and the properties of the distribution (failure rate, mean and standard deviation, or MTTR). Relyence RBD can calculate critical metrics for evaluating system performance: reliability, unreliability, availability, mean unavailability, total downtime, failure frequency, path sets, etc. Moreover, it determines the most efficient computational methods based on system complexity and on the selection of metrics to compute. It includes both the ability to perform analytical calculations and Monte Carlo simulation, allowing in this case to set the number of iterations to perform, the number of failures required to reach steady state, and set the random number generator seed.

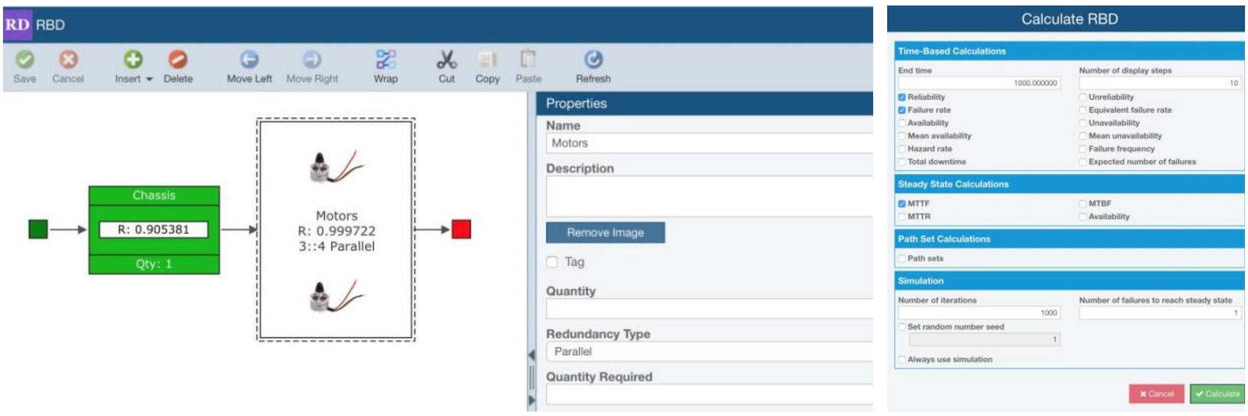


Figure 15 Relyence RBD components detail and options

Through Relyence RBD it is possible to determine and highlight the path sets that result in a successful route from start node to end node, as a mean to evaluate the overall reliability of the system, and to find out areas where improvement may be necessary. Both analytical result outputs and graphical reports are generated and they can be exported to a variety of formats.
The Relyence toolset includes also Relyence FMEA, which supports the most widely accepted FMEA standards:

- Design FMEAs to assess and address failure modes during early or end stage product design;
- Process FMEAs to analyse and maintain process control objectives, including PFDs (Process Flow Diagrams) and PCPs (Process Control Plans);
- FMEA-MSRs, or Monitoring and System Response FMEAs, to supplement DFMEAs and evaluate product failures during customer use
- Custom FMEAs.

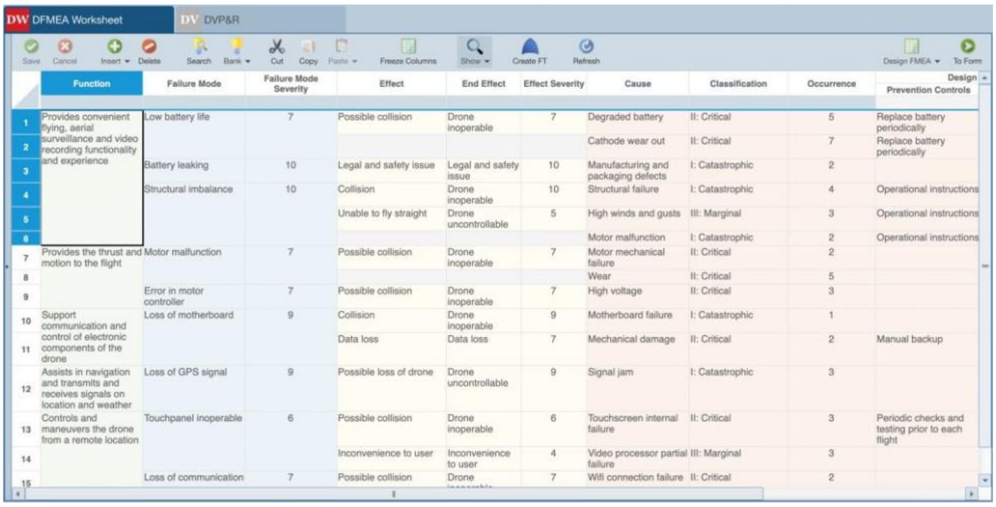


Figure 16 Relyence FMEA

4.3.3 RAM Commander

ALD Group is a company providing reliability engineering, safety and quality software solutions. Among them, RAM Commander is the Reliability and Safety software for reliability of electronic, electro-mechanical and mechanical systems (ADL Group, 2020). RBD module allows performing the functional reliability and availability analysis of systems with variety of reliability distributions, redundant configurations and repair factors.

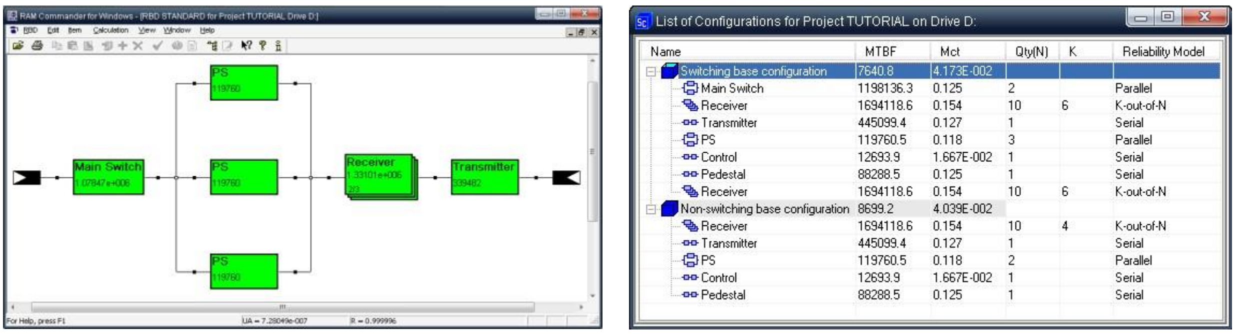


Figure 17 Ram commander RBD

RBD utilizes data defined in other modules (reliability, maintainability, FMECA), and performs either analytical calculation or Monte Carlo simulation depending on the type of system reliability configuration. While the former is used mostly in case of exponential distributions, Monte Carlo simulation is used in the most complex configurations, including standby, partially loaded and active redundancy; full or restricted repair with non-exponential distribution of time-to-repair; analysis under non-steady, transient state; dependent RBD elements.

RAM Commander FMECA software module makes the FMEA process easy and visible: the product tree and the sequence of failure modes, next higher effects and end effects for each item are completely visible in the same window. Extensive use of the FMECA Libraries facilitates the process and contributes to the accuracy of the performed analyses and reports.

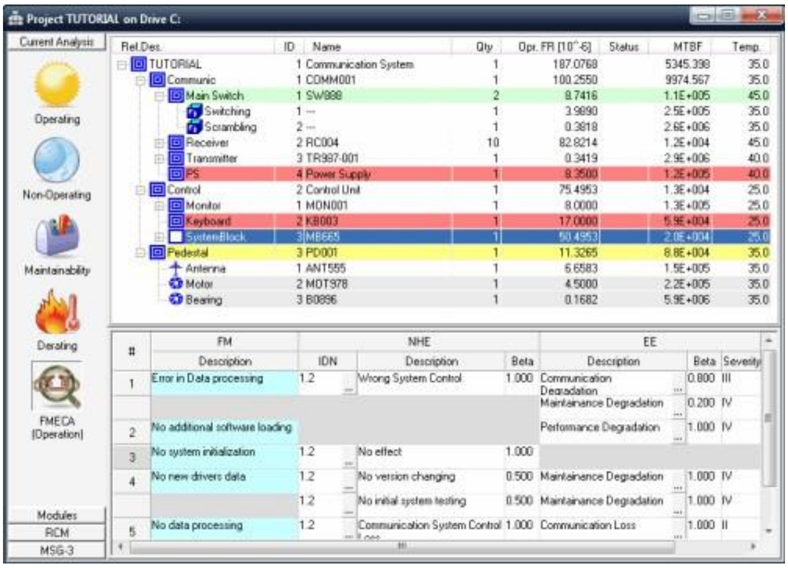


Figure 18 RAM FMEA

Both the FMEA and FMECA can be performed calculating failure mode ratios, conditional probabilities and item criticality numbers. Indeed, the Process & Design FMEA module defines the manner in which a component, subsystem, or system could potentially fail to meet the design intent. The module decomposes the system or process into components or sub-processes. For each functional block, it defines name and function, enters failure mode causes and effects manually or from the libraries. Then it provides full graphical and textual visibility of the potential failure mode, cause and effects.

4.3.4 CARE

CARE is a software developed by BQR and provides multiple types of reliability analysis in a single integrated tool (BQR, 2020).

RBD analysis can model complex redundant systems, with different type of failure distributions, and it can provide a variety of metrics (reliability, availability, downtime, MTBF, MTTR, MTBCF, MTTCF and failure rates). It could import the block tree and components from BQR’s FMECA, while user must provide mission time, block configuration, repair policy, failure distributions, and eventually network diagrams and Markov chain models.

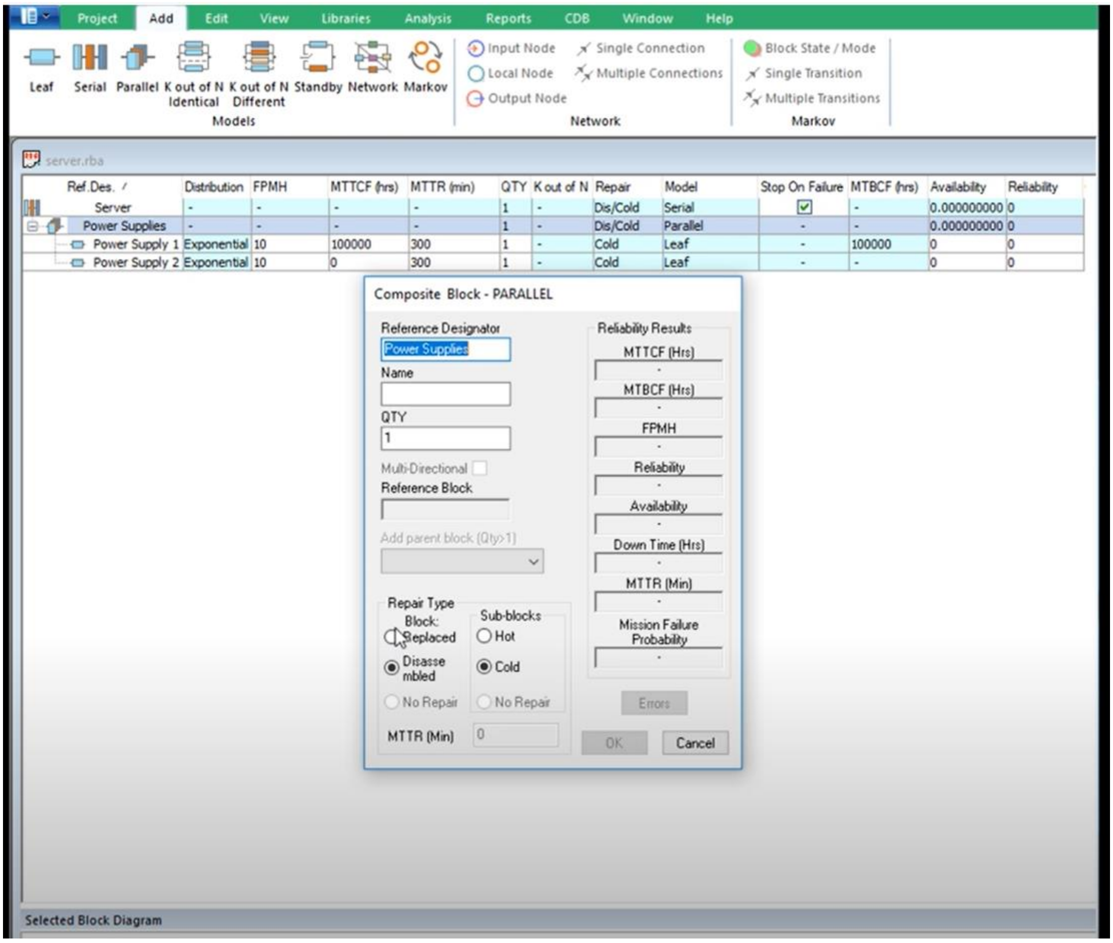


Figure 19 CARE's interface

Then, the software analyses the RBD and calculates all metrics. Both analytic and Monte Carlo simulations are available. The FMEA supported by CARE identifies the most critical failure modes starting from the assignment of component failure rates, failure modes and ratios. Then, user defines Mission time, Severities and Risk Matrix according to required standard, defines functions and dispatch components to functions, define effects of each failure mode up to the system level and severity for each system level failure mode. The software analyses all possible failure paths and calculates the severity and risk for all failure modes and functions, providing standard FMEA and FMECA reports. BQR's FMEA tool supports systems and components FMEA. It is possible to import the block tree of the system and assign components failure rates, failure modes and ratios. Then, the user can define the Mission time, the Risk matrix, which can be customizable, the effects of each failure mode up to system level, and the severity for each system level failure mode.

Failure Mode Effects

ID	Mode name	Next effect			Next effect			End Effect name	Severity
		Block	Mode ...	EP	Block	Mode ...	EP		
6	Stuck Red + Green	Microprocessor	Stuck Red + Green	1	Pressure meter	Red+Green	1	Red+Green	III
7	Stuck No Alarm	Microprocessor	Stuck No Alarm	1	Pressure meter	False Green	1	False Green	I
8	Stuck No Indication	Microprocessor	Stuck No Indication	1	Pressure meter	No Indication	1	No Indication	III
9	Stuck Alarm	Microprocessor	Stuck Alarm	1	Pressure meter	False Red	1	False Red	II

Figure 20 CARE's FMEA

The software analyses all possible failure paths and calculates the severity and risk for all failure modes and functions. The tool gives back a failure modes criticality matrix, as shown in Figure 21, for each severity and criticality group.

Failure Modes Criticality Matrix (Quantity for Internal Causes only)

CRITICALITY		SEVERITY				
Group	**Range**	**V**	**IV**	**III**	**II**	**I**
a	b	d				
A	0.2 - Infinity	0	0	0	0	0
B	0.1 - 0.2	0	0	0	0	0
C	0.01 - 0.1	0	0	3	2	2
D	0.001 - 0.01	0	0	3	1	1
E	0 - 0.001	0	0	1	0	1

Figure 21 CARE's FMEA criticality matrix

4.4. Lessons learnt

Analysing the results of the survey (section 4.2.1), many companies still adopt basic approaches (e.g. reactive maintenance) to the management of failures and maintenance. When asked to evaluate the relevance of the maintenance approaches, the answer with the greatest emphasis was for the reactive approach. Furthermore, 50% of companies do not measure the OEE.

It is therefore quite clear that, although there are companies that adopt very advanced solutions relying on predictive or prescriptive maintenance, many more adopt very obsolete and probably not very effective solutions.

Analysing the tools proposed in section 4.3, they appear to provide numerous features, as summarized in Table 5. However, these tools underestimate the importance of usability and ease of use, in particular for those which have few skills on the topic. Providing innumerable functions and approaches sometimes does not facilitate those looking for tools that can support them in the transition to more advanced maintenance and failure management approaches, but in a gradual and guided way. Finally, they completely lack of solutions to collect data in a structured way.

Analysis / Software	Failure Mode Effect Analysis (FMEA)	Reliability Block Diagram Analysis (RBDA)		
	Supported standards & analysis	System models	Metrics	Additional features
Reliasoft (XFMEA & BlockSim)	**Supported standards:** AIAG & VDA 1st; Edition; SAE J1739; IEC 60812; ISO 14971; VDA-4 (German Automotive industry); MIL-STD-1629A; **Supported analysis:** DFMEA; PFMEA; System FMEA; FMECA.	**Supported configurations:** Simple series; Parallel - total or k-out-of-n redundancy; Load sharing; Standby redundancy - hot, warm or cold; Supported failure and repair distributions: (not specified)	It supports exact computations and/or discrete event simulation (not better specified). **Supported analysis:** System reliability analysis; Identification of critical components; Optimum reliability allocation; System maintainability analysis; System availability analysis; Throughput calculation; Life cycle cost estimation.	• Mirrored blocks • Multi blocks • Sub-diagram options
Relyence (FMEA & RBD)	**Supported standards:** AIAG & VDA; SAE J1739; ARP5580; AIAG FMEA; ISO 26262; Compliance; MIL-STD-1629; MIL-STD-1629A Supported analysis: DFMEAs; PFMEAs; Piece-part FMECAs; FMEA-MSRs; PFDs; PCPs; Custom FMEA.	**Supported configurations:** Series; Parallel; Cold and hot standby (including switch probability and delay), **Supported failure and repair distributions:** Constant Time; Exponential; Gumbel+; Gumbel-; Lognormal; Normal; Rayleigh; Time Independent; Uniform; Weibull.	It supports both analytical calculations and Monte Carlo simulation. **Supported metrics:** Reliability; Failure rate; Availability; Mean availability; Hazard rate; Total downtime; Unreliability; Equivalent failure rate; Unavailability; Mean unavailability; Failure frequency; Expected number of failures; Steady-state MTTF, MTTR, MTBF, availability	• Sub-diagrams linking • Path sets evaluation • Path sets highlighting • Results viewing • Data importing and exporting • Cross-module data integration
RAM Commander (FMECA, Process & Design FMEA modules &	**Supported standards:** AIAG; MIL-STD-1629; GJB 1391; GJB 1392. **Supported analysis:** DFMEA; PFMEA; Potential FMEA; FMECA.	**Supported configurations:** Series; Parallel k-out-of-n with active redundancy (hot); Parallel k-out-of-n with stand-by redundancy (cold); Parallel k-out-of-n partially loaded (warm); Parallel k-out-of-n w/repair, w/o repair,	It supports both analytical calculation and Monte Carlo simulation. **Supported analysis:** Reliability and Availability analysis (not better specified).	• Sub-RBD • k-out-of-n for Sub-RBD • Cross-module data integration

RBD module)		w/restricted repair; Parallel k-out-of-n with Switch. **Supported failure and repair distributions:** Exponential; Normal; Log-Normal; Weibull; Erlang; Uniform.		
CARE (Failure Mode and Effects Analysis & RBD)	**Supported standards:** MIL-STD-882E; MIL-STD-1629A; SAE J1739; IEC 60812; AIAG FMEA-4; EN 20126; EN 50128; EN 50129. **Supported analysis:** FMEA; FMECA (not better specified)	**Supported configurations:** Serial; Parallel; Parallel k-out-of-n; Standby. **Supported failure and repair distributions:** Exponential; Weibull; Log-normal; (the only ones specified)	It supports embedded Markov chain models and Monte Carlo simulation methods. **Supported metric and analysis:** Reliability; Availability; Failure rate; MTBF; MTBCF; MTCF; MTTR; Network reliability; Embedded Markov chain models	Import block tree and failure rates from BQR MTBF, FMECA software or Excel

Table 5 4.3. Tools and software supporting the evaluation of assets and machines status: features sum-up

5. A procedure to support failure and reliability analysis

The proposed procedure aims to support companies in applying failure analysis with a systematic and recurring approach to all processes, supporting the work and allowing the activity to be repeated in the same way. This allows optimizing the use of resources and comparing results. The application of this procedure allows:

- Collecting failure and structure information in order to apply the analysis, understand the causes and identify possible mitigation actions;
- Obtaining a deeper overview and structured information about failure modes and events in order to be able to evaluate if more advanced solutions are needed.

The main characteristics of this procedure are:

- It is agile. Initial assumptions and decisions can be revised in light of new information from the analysis;
- It is applicable to contexts, production systems and processes of a very different nature;
- It allows planning the sequence of activities to be carried out to complete the analysis;
- It supports continuous improvement.

The procedure, which is showed in Figure 22, is divided into 4 different phases, each of them consisting of one or more activities:

- **Planning phase:** definition of the team carrying out the analysis and of members' responsibility; goals setting; definition of investigation perimeter and detail level of the analysis.
- **Informative phase**: identification of production centres, mapping of the production process, related activities and involved systems; identification and classification of systems and their components; failures-related information and data collection; critical observation of processes and systems.
- **Analysis phase**: FMEA, definition of the relation between failures, systems and components, RBD analysis, statistics and metrics analysis.
- **Improvement phase**: implementation of the mitigation actions, maintenance and refurbishment activities definition.

The trigger that springs the (re)application of this procedure can be:

- The company has not yet applied this procedure or any kind of failure and risks analysis such as FMEA, Fault Tree Analysis and/or reliability block diagram;
- The company has carried out relevant modifications in the production system's machines, automation and/or other systems;

- Extraordinary maintenance activity has modified production system's hardware or software;
- Relevant increasing of number of non-conformities due to systems' failures;
- Relevant reduction of system's OEE.

It is also suggested to apply the procedure at least every year to identify new risks and failures and to investigate in more detail those that have been already identified in order to sustain the continuous improvement inside the company.

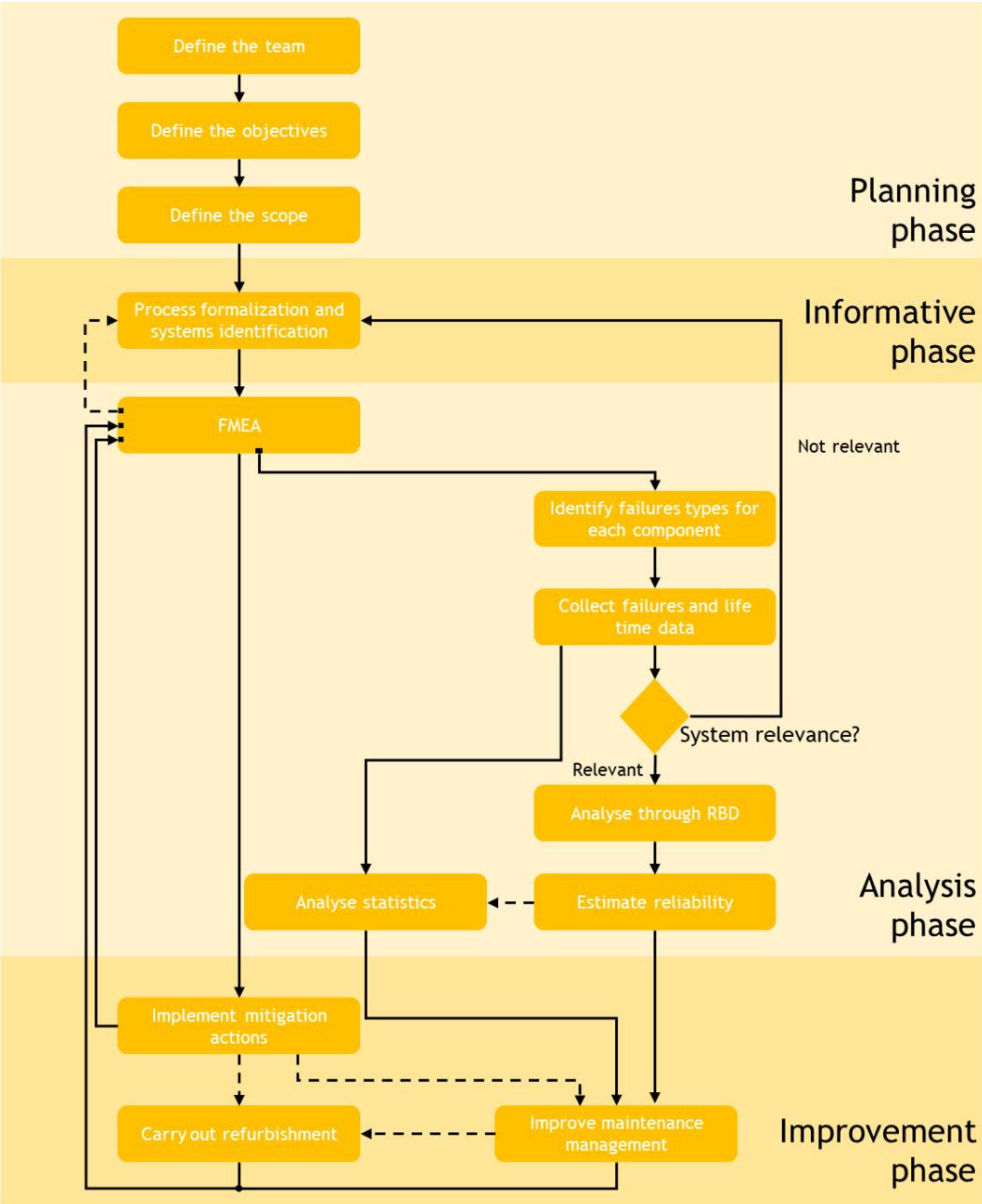


Figure 22 Procedure to support failure and reliability analysis

5.1. Defining the team

As first step, it is fundamental to define the responsible of the whole project that takes care of the application of the overall procedure. Usually such role has to be carried out by a member of the engineering team or by the maintenance manager. However, nothing prevents to assign this responsibility to others.
The project owner has to define the team and the team leader to carry out the various activities in the best possible way. The team must be made up of at least 4 people with multidisciplinary skills and with in-depth knowledge of the production processes and systems involved. The members have to belong to different business areas including Research and Development, Engineering and Quality Control, Operations. The team must also have the capability and possibility to interface with people who work in Production, Quality Assurance and Sales. If the analysed production system is wide, multiple teams and leader could be defined.

5.2. Defining the objectives

The objectives definition is the second activity. It is essential that each team member understands, accepts and shares every single goal. Since this step can form the starting point for setting up the following activities, it is important that each objective has the following characteristics:

- **Relevant** to the context of the project and, therefore, representative of one of the possible solutions;
- **Adequate** with respect to the skills and resources available in the team;
- **Concrete** in such a way that it can be easily translated into practical actions;
- **Reachable** in the established ways and times;
- **Measurable** with specific performance indicators.

For example, a possible objective appears to be: "development and implementation of actions aimed at decreasing by 20% the machines' downtime by the end of the year 2021". It turns out:

- Relevant, because with the tool, the FMEA and RBD analysis, it is possible to identify the failures cause downtimes;
- Adequate, because the requests are aligned with the skills available to the team;
- Concrete, since it can be translated immediately into concrete actions starting from the identification of the machines, and the consequent definition of failures and actions to be carried out;
- Achievable, because within a year it is possible to identify failure modes, apply corrective actions and evaluate the improvements made;
- Measurable, given that, by setting the KPI "machines' downtime", it is possible to understand whether or not the goal has been achieved on schedule.

After the objectives have been defined, the responsible of the project has to take care of informing the internal stakeholders of the company about the defined objectives and the expected timing for their achievement.

5.3. Defining the scope

The correct and clear definition of the scope allows to carry out analyses that are consistent and aligned with the planned objectives.
The scope predominantly concerns two elements: the perimeter of the investigations and analysis and the level of detail defined for their execution. Whenever this step is faced, the team has to question itself if the perimeter of the study influences the reachability of the established objectives and the necessary resources for carrying out the project. Goals and workload are the two main factors that also determine the choice of the detail level that the analysis has to adopt. The analysis of the types and causes of the failures can remain at high level referring to maintenance activities on machinery subsets or can go deeper (e.g. individual components and/or to the code lines of the software). The choice regarding the level of detail involves and, therefore, influences in the same way the granularity of the breakdown of processes and systems, the analysis of the possible effects of failures and the definition of recommended actions and maintenance activities.
Table 6 shows the three categories of the level of detail at which the reliability analysis can be performed. The decision regarding the value must be taken at this stage and maintained throughout the procedure.

Level of detail	Description
3	The analysis is conducted at a high level. The search for causes refers to maintenance activities ordinary and extraordinary performed on machinery without discussing technical aspects.
2	Each activity analyzed corresponds to an activity on the product. The analysis refers to the different subsets of machinery and their interaction, specific components of the means of production and the function they perform during the process.
1	Each activity analyzed corresponds to an activity on the product. The analysis discusses in detail the HW and SW specifications of the production means.

Table 6 Reliability analysis: level of details

Another element to clarify during the scope definition concerns the criterion to be used to assign value to the occurrence index during FMEA. The choice to use the qualitative or quantitative criterion influence the type of information to be collected during FMEA application.
An example of a decision matrix is proposed in Figure 23. It structures the possible decisions regarding the level of detail and the criterion to assign the value to the occurrence parameter using the specificity of the objectives and the financial significance of the analysed production system.

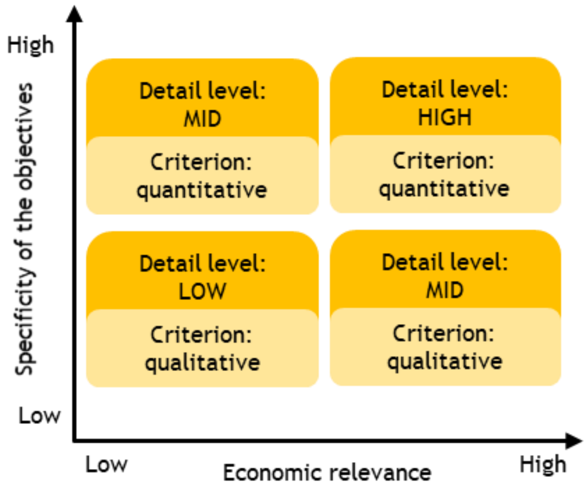


Figure 23 Example of decisions regarding the level of detail and probability of occurrence index

5.4. Process formalisation and systems identification

This step aims at identifying activities, systems and components and, as part of the informative phase, at the collection, the information necessary to carry out the FMEA.
The information sources available in a company are multiple and of different nature (e.g. report of previous projects, technical specifications of the machinery and systems).
The 4 activities that compose this phase are:

- **Identification of production centers, mapping of the production process, related activities and involved systems**. The representation and formalisation of the production process is always a fundamental step to carry out any kind of analysis in a production system. To this end, different formalism and standard can be adopted such as Flow Charts, BPMN, IDEF0, etc.
- **Identification and classification of systems, including all machines, equipment, tools, etc. and their components**. During this activity, a complete list of the systems in the shop floor has to be defined, reporting at least name and installation date. Moreover, each system has to be classified according to the parameters reported in Table 7, namely relevance, replaceability and maintainability. The product of the three parameters allows calculating a System Criticality Index (SCI), which in turn allows establishing the criticality for the system, and, as consequence of its components and failures, for the company.
- **Failures-related information and data collection**. All the relevant information related with failures has to be collected from the different data sources. Some examples of data sources include: report of previous projects, technical specifications of the machineries and systems, documents containing consultants' analysis, documents physically or digitally stored in the information system

reporting failures, maintenance and revisions histories, process activity description, articles produced, internal procedures, setup activities for regulation and control of the systems, etc.).

- **Critical observation of processes and systems (e.g. through questionnaires to operators).** This activity has to be carried out directly next to the workstations and systems in operation, allowing to immediately highlight the failure modes, preliminary identified after having consulted the collected documents. The dialogue with operators, maintainers and other figures working in the shop floor is the real added value of this activity. Thanks to the training received and experience matured on a daily basis, operators are a valuable source of information in support of the failure analysis. The recommended approach initially involves questioning the production manager about the critical aspects of the process and system in analysis, defining the main causes of rejects, the type of controls carried out during the activities and the reasons for the extraordinary maintenance interventions. After having structured the information collected, it is necessary to query the operators. It is recommended to carry out at least 3 direct interviews by asking each worker the same questions. Alternatively, a focus group can be used, in which the simultaneous presence of at least three operators is expected. However, single interviews are more effective, because they allow collecting a greater quantity of information, expressed from different points of view. The questions have to be focused on the failure modes, the possible causes and the components that are involved. Possible examples of questions could be:
  - What happens if the ribbon is not loaded or has the wrong orientation?
  - What is the cause of these marks on the semi-finished product?
  - Could the product get dirty with oil?

  Some aspects affect the reliability of the information provided by operators, such as:
  - Partial or incomplete view of the production process;
  - Lack of experience in certain tasks;
  - Fear of possible repercussions;
  - Friction with work colleagues.

  A second meeting with the relevant figure acting in the production system (e.g. production managers, team leaders, technicians, etc.) is therefore suggested.

Parameter	Description
Relevance	This parameter classifies the relevance of a system, in particular how it contributes to the value creation. To evaluate this parameter, different methods can be adopted, such as AHP (focus group or single interviews comparing one to one each system) and/or Pareto/ABC analysis (on value added, profit,

	revenue or volume). The scale to evaluate this parameter is composed by 4 grades: • Fundamental (4) • Relevant (3) • Not essential (2) • Irrelevant (1)
Replaceability	This parameter classifies the replaceability of a system, in particular if the activities it is used for are, or can be, performed by other systems. To evaluate this parameter, different methods can be adopted, such as AHP (focus group or single interviews comparing one to one each system) and/or technological/process constraints analysis. The scale to evaluate this parameter is composed by 4 grades: • Irreplaceable (4) • Replaceable with a high expenditure of resources (3) • Replaceable with a small expenditure of resources (2) • Easily replaceable/redundant (1)
Maintainability	This parameter classifies the maintainability of a system, in terms of costs and effectiveness. To evaluate this parameter, different methods can be adopted, such as AHP (focus group or single interviews comparing one to one each system) or costs and MTTR analysis. The scale to evaluate this parameter is composed by 4 grades: • Low maintainability (high costs and high MTTR) (4) • Medium maintainability (high costs or high MTTR) (3) • High maintainability (not relevant costs or not relevant MTTR) (2) • Excellent maintainability (not relevant costs and MTTR) (1)

Table 7 Systems' classification parameters

5.5. Performing FMEA

The FMEA is the first analysis to be performed in the procedure. At this point, the team shall be in possession of:

- A formal representation of the production process;
- A list of classified systems and their components;
- All the available information related with process and system failures collected from documents and from workers interviews.

To start the FMEA, the team has to propose, discuss and confirm each possible failure mode, their respective causes and effects, and, finally, assess the presence of active controls on the process, as represented in Figure 24.

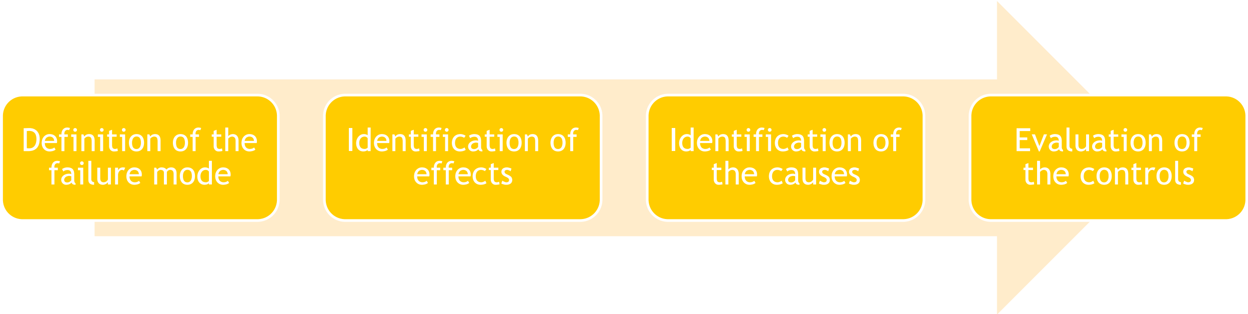


Figure 24 FMEA-first activities

These activities have to be performed during a meeting, where one or more team members explain all the failure modes that have been identified analysing the available information. Each member, based on his/her skills, is required to evaluate each failures, expressing his/her opinion regarding the need to deepen it and if include it in the FMEA or discard.

After having identified all failure modes, the focus of the meeting has to move on to the discussion of all possible effects and causes. Possible effects have to be identified considering the occurrence on:

- Production system;
- Product;
- Customer.

It is important that, during the entire course of the meeting, the FMEA template, represented in Table 11, is visible to all participants so that they can actively intervene and help the team leader in completing it.

After having identified all the failures, during the same meeting or later, the team has to evaluate failures indexes. It consists in assigning values to severity, occurrence and detectability in order to allow the calculation of the Risk Priority Number (RPN).

The use of a common evaluation scale fixed a priori for each index allows the comparison between each failure. These have to be maintained throughout the course of the whole analysis. Examples of rating scales are shown in Table 8, Table 9 and Table 10. Each company applying FMEA can define its scales, depending on the sector, the processes, the products, the policies that characterize it.

Table 8 shows an example of scale for the severity index from a company producing medical devices. As with the failure effects, also severity has to be evaluated for production system, product and customer. In case of different severity rank for the three items, the highest has to be considered.

Rank	Description		
	Production system	Product	Customer
High (5)	Production system destruction	Complete functionalities loss and damages to the environment with which it is in contact	Death
Relevant (4)	Relevant damages to the system	Complete functionalities loss	Permanent disability or irremediable injury

Medium (3)	Production process stop and/or damages to the production system	Product does not perform the expected functions and/or it requires repairs.	Injury requiring a surgical intervention
Low (2)	Increased cycle time or/and more resources needed to obtain the same output	Performance reduction	Temporary injury not requiring surgical intervention
Tolerable (1)	Not relevant impact on the production system and production process	Effect on packaging and/or on aesthetic, but not on functionalities	Temporary discomfort, and/or disease

Table 8 Severity scale: example from a company producing medical devices

Table 9 shows an example of scale for the occurrence index from a company producing medical devices. There are many aspects that can be considered for assigning the occurrence rank. The decision on which approach to adopt (quantitative vs qualitative) has to be defined during “Define the scope” step. The use of the qualitative approach certainly makes the results more reliable and truthful, but involves a greater expenditure of resources and the need for more information, in particular if any history of failure is already available in the company.

Rank	Description	
	Qualitative	Quantitative (% of produced parts)
Frequent (5)	The event is frequent and the failure mode is almost constant over time	≥ 5%
Probable (4)	The event happens likely	< 5% and ≥ 1%
Occasional (3)	The event is infrequent	< 1% and ≥ 0.1%
Remote (2)	Event hardly occurs	< 0.1% and ≥ 0.001%
Improbable (1)	Event is almost impossible	≤ 0.001%

Table 9 Occurrence scale: example from a company producing medical devices

Table 10 shows an example of scale for the detection index from a company producing medical devices. As with the severity index, in this case the only available approach is qualitative. However, two different methods to assign the value to the parameter exist:

- Evaluate if the controls in the process are able to detect failure events and their causes;
- Evaluate whether the controls in the process are able to detect the effects triggered by failure modes.

Rank	Description
Improbable (5)	There is no opportunity to identify the failure event. There are not controls during the process.
Low (4)	The opportunities to identify the failure event are few and the controls are only visual (e.g. comparison with a standard model)

Moderate (3)	The opportunities to identify the failure event are moderate and controls with measurement are carried out and the decision is made based on a numerical result documented through a validated test.
High (2)	The opportunities to identify the failure fault are high, checks are carried out thanks to measurements made by the operator (checks set-up is valid from a statistical point of view).
Almost certain (1)	The detection of the failure event is almost certain (it is the case of a validated and automatic control on 100% of the production)

Table 10 Detection scale: example from a company producing medical devices

RPN is calculated by multiplying severity, occurrence and detectability. RPN has to be used to compare failures. However, thresholds should not be assigned due to two reasons:

- Relative Risk is not always represented by the RPN value;
- Poor behaviour by the FMEA team trying to get below the specified RPN thresholds.

To complete FMEA, the short-term actions adopted to overcome the failure without solving it have to be identified, in order to simplify the definition of the mitigation actions that can:

- Eliminate the failure;
- Reduce the occurrence and/or severity;
- Improve the detectability.

Finally, the team has to define a plan including the defined mitigation actions, deadlines and responsibilities, in compliancy with RPN, company's priorities and resource availability.

Field name	Description
Failure ID	To facilitate communication between different stakeholders, an ID has to be assigned to each failure type. The company can define its method to define its failure IDs.
Process/activity analysed	Name of the process/activity used in the formal representation of the production process. Each process/activity name has to be clear and easily distinguishable from the others.
Function	This field has to provide details of the process/activity and, in case, the role of the machine and component. E.g. the operator drills the sub-assembly using the vertical drill.
Failure mode	c
Potential effect of the failure	Potential effect that can be generated by the failure on product, production system or customers.
Severity	Value of the severity index assigned to the failure according to the defined scale and criteria (e.g. see Table 8)

Rationale for the choice of the relevance value	Reasons behind the selection of the severity index value.
Potential causes of the failure type	List all the potential causes that can generate the analysed failure mode.
Cause type selection	Manpower; System, equipment or machine; Method; Material; Measurement; Environment; Other
System	Name of the machine which involves the failure. This field could be empty.
Component	Name of the component which involves the failure. This field could be empty.
Current controls	List all the controls that are in place that can identifies the failure mode.
Occurrence before mitigation actions	Value of the occurrence index assigned to the failure according to the defined scale and criteria (e.g. see Table 9)
Rationale for the choice of the occurrence value	Reasons behind the selection of the occurrence index value.
Detectability before mitigation actions	Value of the detectability index assigned to the failure according to the defined scale and criteria (e.g. see Table 10).
Rationale for the choice of the detectability value	Reasons behind the selection of the detectability index value.
Risk Priority Number (RPN)	Product between the three indexes which represent the criticality of the failure mode.
Short-term actions	List of actions currently adopted to overcome the failure.
Mitigation actions	List of actions that will be adopted to mitigate the failure reducing one or more parameters. Each action has to include the description, the responsibilities and the deadlines.

Table 11 FMEA template

5.6. Mitigation actions

Mitigation actions have to be implemented according with a defined plan which includes responsibilities: who is going to develop the activities, who is going to control the effective implementation and the achieved results. Each mitigation action has to be documented and, if unsuccessful, another one has to be determined. After actions have been taken, the new (re-ranked) RPN should be compared with the original RPN. If the RPN is still too high after action, a new action has to be defined and implemented. This is repeated each time until an acceptable level of risk has been obtained.

5.7. Identify failures types for each component

Failures causes can be classified in Manpower; System, equipment or machine; Method; Material; Measurement; Environment; Other. Those classified System, equipment or machine can be subjected to further analysis, in case the needed resources are motivated by the relevance of the machine, calculated as (see Table 7):

$$System's\ criticality = Relevance * Repleacability * \ Maintainability$$

To this end, each failure mode with a cause classified as "System, equipment or machine" has to be associated to a system, and if possible, with one of its component.

5.8. Collecting failures and life-time data

FMEA and the identification of the relation between components and failures allow to have a structured and detailed overview of failures due to company's systems.
If systems, components and failures types have been structured correctly, the company has to define one or more procedure to collect failures data. This has to support the monitoring of the occurrences of failures, in order to have a deeper details on failures (e.g. statistics, etc.) on the process and have more detailed information for future application of the FMEA. These procedures have to be adopted in order to collect data on failures, such as:

- Which process failed
- Which failure mode occurred
- When failure occurred;
- How the failure has been detected;
- Which short term actions has been adopted to overcome the failure;
- If the component has been replaced;
- Time to repair;
- Other relevant elements if necessary.

This data has to be used to collect failures history in order to allow statistics and analysis introduced in the following steps.

5.9. Analysing through RBD

As described in Figure 22, before applying this step, it is necessary to evaluate the relevance of each system. RBD analysis requires relevant effort in terms of time and resources. For this reason, it is suggested to consider the system relevance calculated through the formula proposed in 5.7. In some cases, detailed analysis such as RBD on systems that are not critical for the production system, could be avoided being the benefits lower than the required efforts.
After having identified which systems have to be included for the RBD analysis, it is necessary to model them according to the RDB standards and set-up all the parameters necessary to a complete analysis. To model and set-up the RBD is necessary to:

1. Identify and model the structure of each system by defining the logical connection and the reliability structure of its components.

2. Define how to estimate systems and components failures distributions (e.g. vendors function and parameters, failures distribution fitting, etc.).

To perform RBD analysis, many tools can be adopted, such as the one proposed in this document, those proposed in section 4.3, or Matlab.

5.10. Estimating reliability

Thanks to the modelling and set-up of the RBD, it is necessary to calculate the reliability of each system in order to evaluate the current and the future status of the production system. This estimation can support the identification of the most critical systems and components in order to plan and implement mitigation actions, maintenance and refurbishment activities.

5.11. Analysing statistics

This step has to be used to analyse statistics and metrics exploiting collected data and RBD analysis. Metrics indication could address the management team in the decision about the improvements to bring in the adopted maintenance strategies.

5.12. Implementing mitigation actions

The defined mitigation actions must be applied with the aim of reducing the risk of failures. The team provided the description of the steps to be carried out for each single mitigation action. It is important also to define responsibility: someone has to follow up on the recommendations to determine if they have been addressed adequately and/or if they are in need of updating. After the action has been taken, the effective date or completion date with a brief description of the action should be entered.
After the actions are incorporated in the system, the related consequences should be reviewed to address for other actions or to configure them as standard in the system process.

5.13. Improving maintenance management

At the light of the results of previous steps, the team, together with the maintenance manager, has to evaluate the actual maintenance strategies, with the aim of identifying gaps and/or eventual means to introduce changes. Whether possible, at least in case of companies for which maintenance is not a brand asset, an advance maintenance strategy should be adopted, moving towards full re-use of equipment in manufacturing.

5.14. Performing refurbishment

The ultimate goal of the analysis is to evaluate whether it is possible to save valuable resources by reusing and upgrading equipment instead of discarding them. The company can indeed assess new optimal strategies for refurbishment.

6. RECLAIM Reliability Analysis Tool (RAT)

The RECLAIM Reliability Analysis Tool (RAT) has been developed in order to partially cover the gaps presented in section in 4.4. The RECLAIM Reliability Analysis aims to support companies, in particular SMEs, which have lack of competences and no or weak approaches to maintenance and failure management, in starting their journey toward the digitalisation of assets health and maintenance management.
The functions that the tool provide are not innovative or complex. Also non-expert users can use it. The main functionalities that it provides are:

- **Collecting structured failure data;**
- **Applying FMEA through a guided procedure;**
- **Applying RBD analysis.**

The following sections provide a detailed description of the tool, including specifications, architecture and deployment approaches.
Tool is accessible at http://isteps-sps-01.supsi.ch/reclaim/login.

6.1. High level specifications

From the needs and requirements highlighted in section 0, the following specifications have been defined to guide the design and development of the RECLAIM Reliability Analysis Tool.
This tool targets SMEs and companies which do not have experience with failure management methods and digital solutions. Therefore, it has been designed to support two different kinds of users:

- **Newbie**: users who need to follow guided procedures to use the tool and apply the methodologies that it implements.
- **Experts**: users who are comfortable with the methodologies on which the tool relies on and do not require guided procedures to use it.

The tool has been developed as a web-application capable of supporting different users. As reported by Figure 26, the tool has to provide the following functionalities:

- **Managing multiple systems:** user has to be able to define and describe an arbitrary number of systems.
- **Managing multiple components for each system:** user has to be able to characterize a system with a series of components.
- **Modelling systems according to RBD standard:** components are the foundations of the reliability block diagram. Once the user has defined the components, he/she has to be able to create and model a RBD diagram.
- **Managing multiple failures for each component:** the user has to be able to define multiple failure types. Each failure has to be associated to a system and

to a component. Moreover, the tool has to support the FMEA procedure on each failure mode.

- **Collecting failure events:** user has to be able to create failure events specifying multiple information to support statistics and metrics calculation.
- **Supporting FMEA:** user has to be able to apply FMEA through a specific guided procedure supporting non-expert user in applying this method.
- **Supporting RBD analysis:** once the user has accomplished the design of the diagram, if all the necessary data are provided, he/she has to be able to calculate failure metrics of the system. The tool has specific guided procedures and a dedicated modeller in order to support non-expert user in applying RBD analysis.
- **Supporting failures distribution estimation:** from failure events the tool is capable to estimate failure event distribution.

Since the tool targets multiple companies, an authentication procedure has to be included. This allows to have personal users and keep work contained and separated from each other. Moreover, since inside the same company, the tool could have multiple users, there is the need to have to two user categories:

- Basic: it can create arbitrary systems and manage all the aspects of them.
- Admin: it can manage all the aspects of the tool, including other users systems.

6.1.1 Use case

Figure 25 shows a use case diagram for the whole platform. The tool has two main actors: User and Administrator.

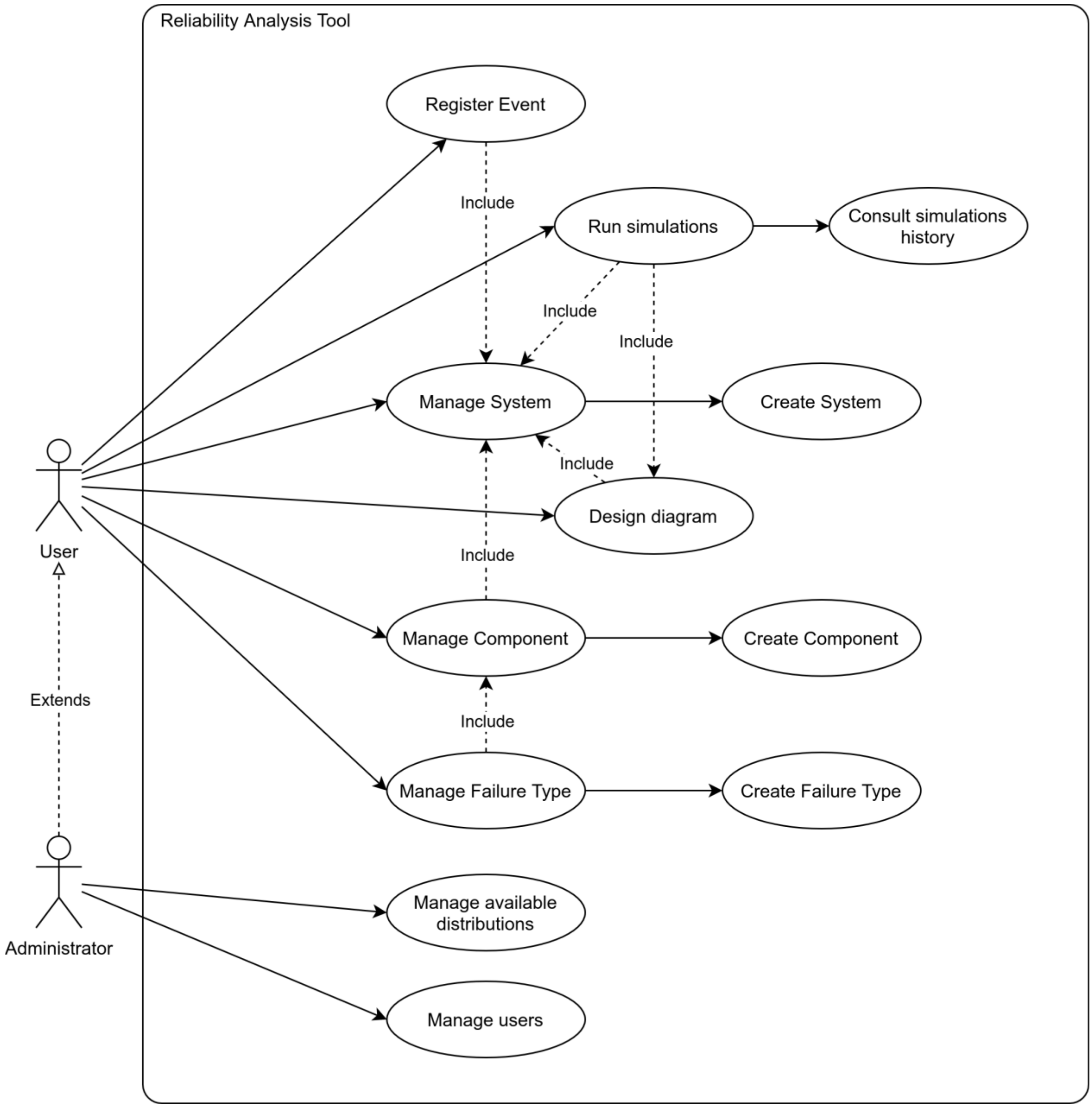


Figure 25 Use-case diagram

The following tables will illustrate in a more detailed manner all the use cases showed on the picture above, the *Actor* is notated as "A", meanwhile the *System* is identified as "S".

Use case	**Create system**
Actors	User, Administrator
Overview	The user creates a new system
Requirements	User authenticated

Goals	Create a new system
Driving actions	1) A: Ask to create a new system 2) S: Ask the actor for the parameters of the new system 3) A: Input the parameters 4) S: Check the validity of the parameters a) A: Correct the wrong inputs b) S: Go to 4) 5) S: Save the new system and show its detail to the user

Use case	Manage system
Actors	User, Administrator
Overview	The user selects a personal system to work on
Requirements	➢ User authenticated
Goals	Manage an already created system
Driving actions	1) A: Asks for personal systems 2) S: Show list of personal systems 3) A: Select a system 4) S: Show the selected system details

Use case	Create component
Actors	User, Administrator
Overview	The user creates a new component
Requirements	➢ User authenticated ➢ System selected
Goals	Create a new component under the selected system
Driving actions	1) A: Asks to create a new component 2) S: Ask the actor to fill the inputs parameters 3) A: Fill the parameters

	4) S: Validate the actor input a) A: Correct the wrong input b) S: Go to 4) 5) S: Show the new Component

Use case	Manage component
Actors	User, Administrator
Overview	The user can manage a previously created component
Requirements	➢ User authenticated ➢ System selected
Goals	Enable the user to manage a component
Driving actions	1) A: Ask the system the list of components 2) S: Show all the available components into the selected system 3) A: Select a component 4) S: Show the component detail page

Use case	Create Failure Type
Actors	User, Administrator
Overview	A component needs to have at least one failure type to let the user create a failure related to the latter. This case describes how the user can create a new failure type.
Requirements	➢ User authenticated ➢ System selected ➢ Component selected
Goals	Enable the user to create a new failure type
Driving actions	1) A: Asks to create a new failure type 2) S: Ask the actor to fill the inputs parameters 3) A: Fill the parameters 4) S: Validate the actor input

	a) A: Correct the wrong input b) S: Go to 4) 5) S: Show the new failure type

Use case	Design diagram
Actors	User, Administrator
Overview	The user needs to model a RBD, this use case supports the latter in this operation
Requirements	• User authenticated • System selected • At least one component created
Goals	Model the system reliability block diagram
Driving actions	1) A: Add a component to the diagram surface 2) A: If necessary create a relationship 3) A: Go to 1)

Use case	Run Simulations
Actors	User, Administrator
Overview	Allow the user to execute a simulation on the reliability block diagram (s)he has designed, and obtain an output
Requirements	• User authenticated • System selected • At least two components on the diagram
Goals	Compute the reliability analysis on the RBD
Driving actions	1) A: Ask to run a simulation 2) S: Ask the user for the mission duration 3) A: Provide the mission duration 4) S: Compute each component reliability 5) S: Compute the RBD reliabiltiy 6) S: Display the result to the Actor

Use case	Consult simulations history
Actors	User, Administrator
Overview	Allow the user to consult past simulations
Requirements	➢ User authenticated ➢ System selected ➢ At least one simulation has been performed
Goals	Give the ability to the user to consult past simulation results
Driving actions	1) A: Ask to get the simulations history 2) S: Shows the past simulations in chronological order 3) A: Select a simulation 4) S: Shows the simulation detail to the actor

Use case	Register event
Actors	User, Administrator
Overview	Allow the user to consult past simulations
Requirements	➢ User authenticated ➢ System selected ➢ At least one simulation has been performed
Goals	Give the ability to the user to consult past simulation results
Driving actions	5) A: Ask to get the simulations history 6) S: Shows the past simulations in chronological order 7) A: Select a simulation 8) S: Shows the simulation detail to the actor

6.1.2 High-level functional specifications

Figure 26 resumes the main functionalities of the tool, defined from the procedure presented in section 5.

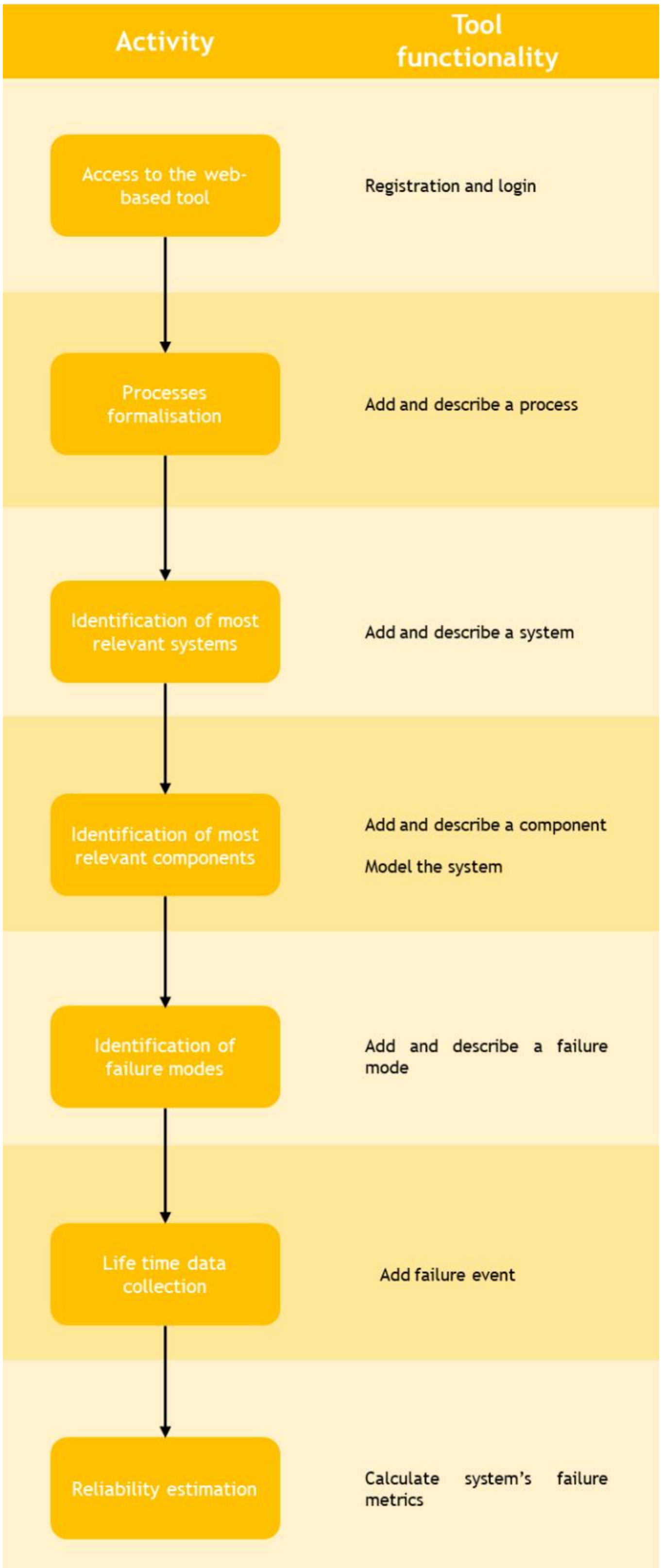


Figure 26 RECLAIM RAT functional specifications

6.2. System design

Development environment, architecture and data model are presented in this section, providing a sum-up of the main elements composing the RECLAIM RAT.

6.2.1 Environment

This section describes the environment and the technological choices implemented.
The project code has been structured into four parts:

- Data model - an abstract model that organizes elements of data and standardizes how they relate to one another.
- Front-end - the presentation layer of the platform.
- Back-end - data access layer of the platform.
- Plugins - plug and play distributions plugin for an easy extension of the platform.

JetBrains (JetBrains, 2020) has been used to manage the development of this project. Each part of the source code has been pushed into a git repository in order to allow a secure versioning management. The data model has been also published on a maven repository to allow an easier dependencies management.

6.2.1.1 Data model

The data model has been written in Java and it formalizes the data structure of the whole tool. The choice of using Java as programming language resides in the fact that also the back end has been written in Java. This allows an easier integration between the two parts.
More details about the data model are provided in section 6.2.3.
The link to the git repository is https://git.jetbrains.space/sps/rat/rat_data-model.git

6.2.1.2 Front end

For what concerns the front-end, 2 possible solutions have been evaluated:

- **Javascript + CSS + HTML based Platform:** developers can write pure Javascript or use some framework like AngularJS, React or something less complex like jQuery. The same thing for the styling part, bootstrap comes in handy to apply quickly a pleasant style to a web page. By using this kind of technology will also require some toolkit to have a quick and functional web-application like Webpack, which is a toolkit used to compile JavaScript components into a single, loadable bundle or Babel, which allows developers to write JavaScript code by using ES6 and compile it into ES5 to run in the browser.
- **Java based Framework**: the GUIs are directly written on java thanks to frameworks like GWT and VAADIN. The main difference between GWT and VAADIN is the fact that the purpose of GWT is just for the presentation layer; in fact, the compiled java is translated to the standard JS+HTML+CSS. VAADIN, on the other hand, operates at server side, so it generates pages directly at server side.

Figure 27 shows the interest over time (estimation based on google searches) about the web development technologies: React and Angular are the dominating technologies. It is therefore advisable to stick with the last and most spread technologies to have most resources possible on the topic.

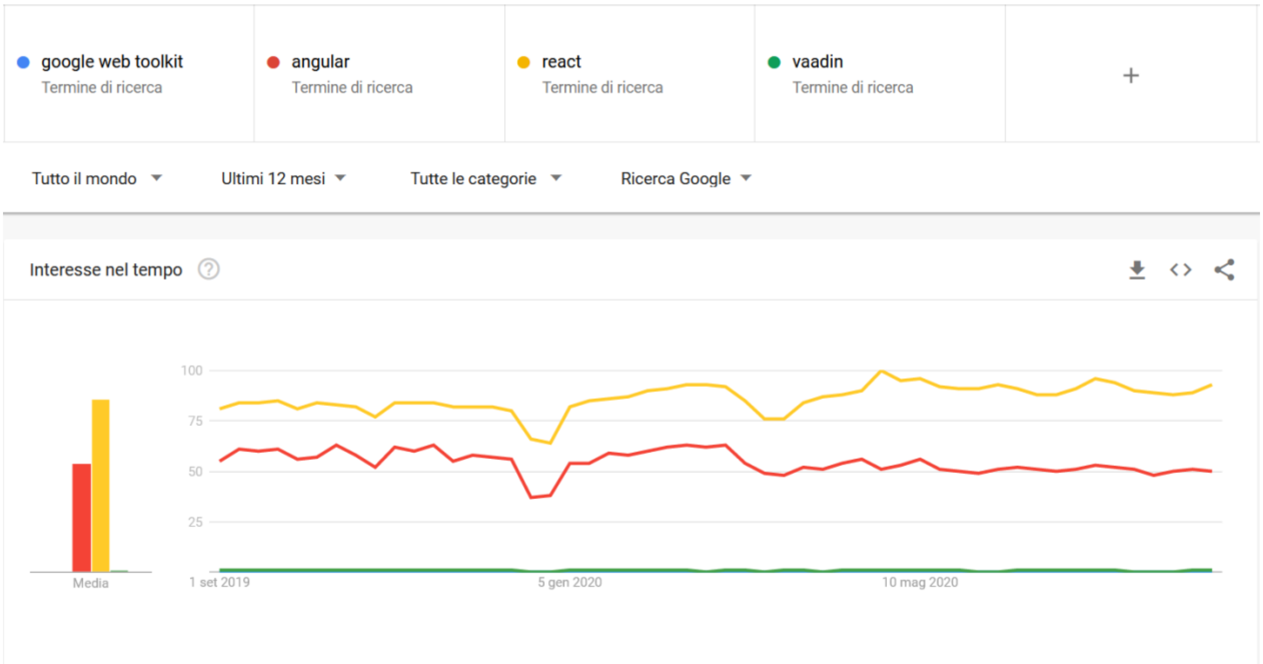
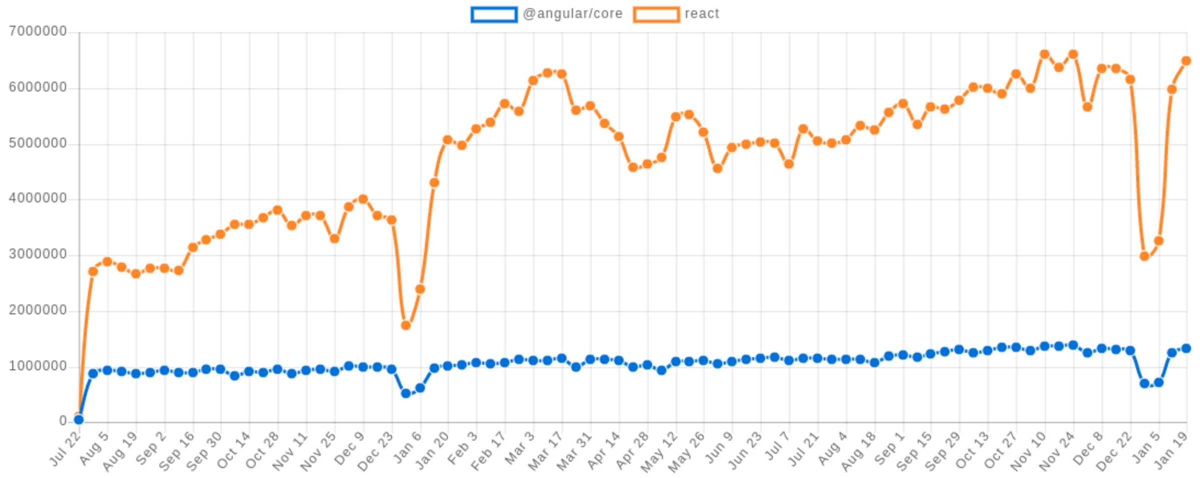


Figure 27 Front-end technologies: interest over time (September 2019 to August 2020)

According to Figure 28, it seems that Angular is better for large-scale projects, while react is the most downloaded one from the NPM package manager, and it is growing very quickly.

Downloads in past 2 Years



Stats

	stars	forks	issues ⚠	updated 🛠	created	size
@angular/core	56649	15586	3400	Jan 22, 2020	Sep 18, 2014	minzipped size 85.7 KB
react	142713	27371	616	Jan 22, 2020	May 24, 2013	minzipped size 2.6 KB

Figure 28 Downloads stats (in past 2 years): React vs Angular

After the proposed investigation, the choice has fallen on a JavaScript front-end supported by React as framework for the following reasons:

- React is easy to learn and RAT developers have already a good experience with it. It allows to easily and fast build user-interfaces easily using the declarative components.
- React has a very strong community support (i.e. searching on google React tutorial, it gives more than 195’000’000 results; searching VAADIN on google gives 155’000 results).

6.2.1.3 Back-end

To develop the RECLAIM back-end, almost every programming language could be adopted. Since the team working on the project is most familiar with Java, it has be selected as programming language for the back-end.
The two possibilities evaluated have been the use of Vanilla Java with some dedicated libraries, or the use of a specific Java framework. Using plain Java give the developer full control on what is happening behind the contains and allow a more efficient resource management, since only what is needed is implemented. The power of using a Java framework is that it allows the developer to focus on the core functionalities of the application instead of (re)implementing the basics functionalities, like the http request handling, making the database connectors or also handling the exceptions. There are many different frameworks to build robust and efficient back-ends working

with Java. Figure 29 shows a graph representing the interest over time of the most adopted frameworks. *Vaadin* and *GWT* have been included because they can be used independently on the back-end side.

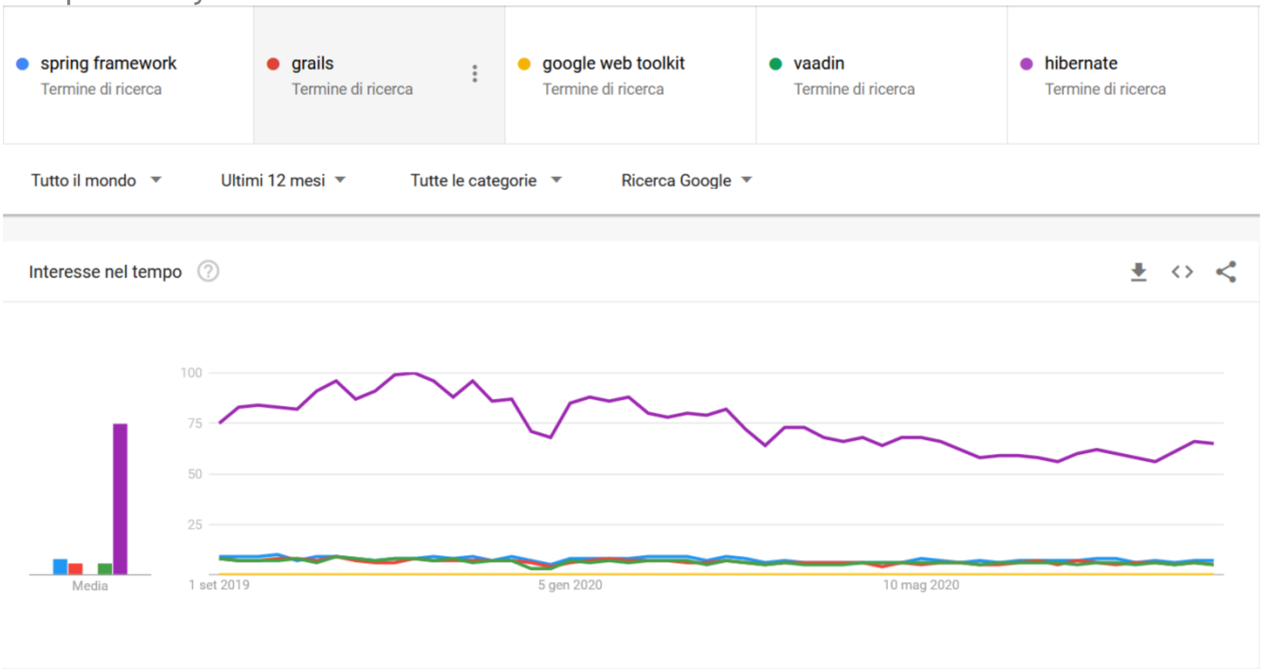


Figure 29 Java frameworks: interest over time (September 2019 to August 2020)

6.2.1.4 Plugin

The software engineering literature provides some variants to manage a plugin infrastructure in Java such as OSGi, pf4j or a mere "manual" management. A "manual" management means loading all interested classes (most probably those implementing a specific interface) from a jar stored on the file system. OSGi instead, would be a valid alternative because it represents the main framework for a plugin-oriented architecture. In this case, although the architecture would be ready and the learning curve is really steep, such a structure would be excessive.
An interesting solution, based on a (micro) framework meant to manage plugins in an easier way, is **pf4j,** which provides everything needed for having a plugin infrastructure reliable with few lines of code. Moreover, it seems to be fairly well maintained since the repository is constantly updated.

6.2.1.5 Database

Almost every RDBMS potentially fit with the RAT's scope, as they are almost all equivalent in performance and use. The real constraint is the license aspect, since the main objective is to support companies, the database must be under a fully open license even for commercial use. PostgreSQL is a great candidate that implements a custom license that allows a free use to everyone. Another aspect that must be considered is security, since the platform is dealing with a lot of personal information, PostgreSQL result being the most secure.

6.2.2 Architecture

A close look at the internal components of the software module can be provided taking into account the internal architecture, which is presented in Figure 30 using a UML component diagram. The software is represented as a big block containing all the necessary components to handle both functional and non-functional requirements.

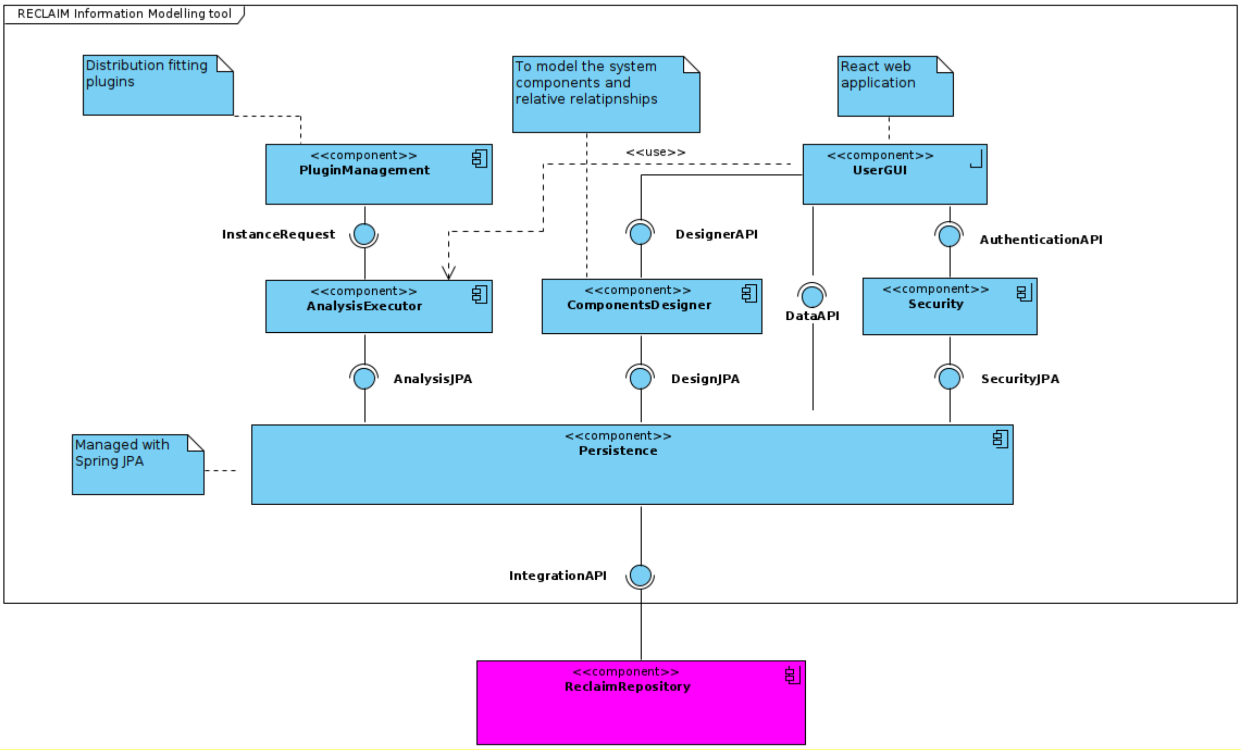


Figure 30: Reliability Analysis Tool – Architecture

Each component is presented and detailed in the following sections.

6.2.2.1 Plugin Management

Since it may be required to have custom distribution fitting functions, a plugin approach has been chosen. The component will handle all the aspects of managing the various plugins installed and to be installed on the modelling platform.

6.2.2.2 User GUI

This component represents the whole GUI that the user will be presented with. Not only the graphic aspect but also the retrieval of information to be displayed on the latter.

6.2.2.3 Analysis Executor

Component in charge of executing the analysis upon user request. The executor will initialize the required function from the Plugin Management component and save the results on database, where they can be consulted by the user or stored into the RECLAIM repository.

6.2.2.4 Components Designer

This component will handle the design of the system structure, allowing the user to design a flow diagram that represents the components in the system.

6.2.2.5 Security

This component is in charge of authenticating the users and allowing them to operate only on their systems.

6.2.2.6 Persistence

The persistence component is in charge of handling all the aspects of the data storage of the platform. An embedded database like H2 or a server database like PostgreSQL will be used. The integration with the database and the rest of the application will likely be implemented using the Spring JPA interfaces.

6.2.3 Data model

The data model is used to formalize a set of classes and interfaces to represent the data structured used within the code. The model is shared also to the plugins in order to have all the components aligned with the project data model.

Figure 31 represents the data model. The *DiagramElement* object represents an element of the diagram. All the sub-types are used to represent the various entities of the diagram. The *ComponentNode* is in charge of establishing a relation between the diagram and the system nodes. They are been kept in different object to better structure and organize the data inside the tool.

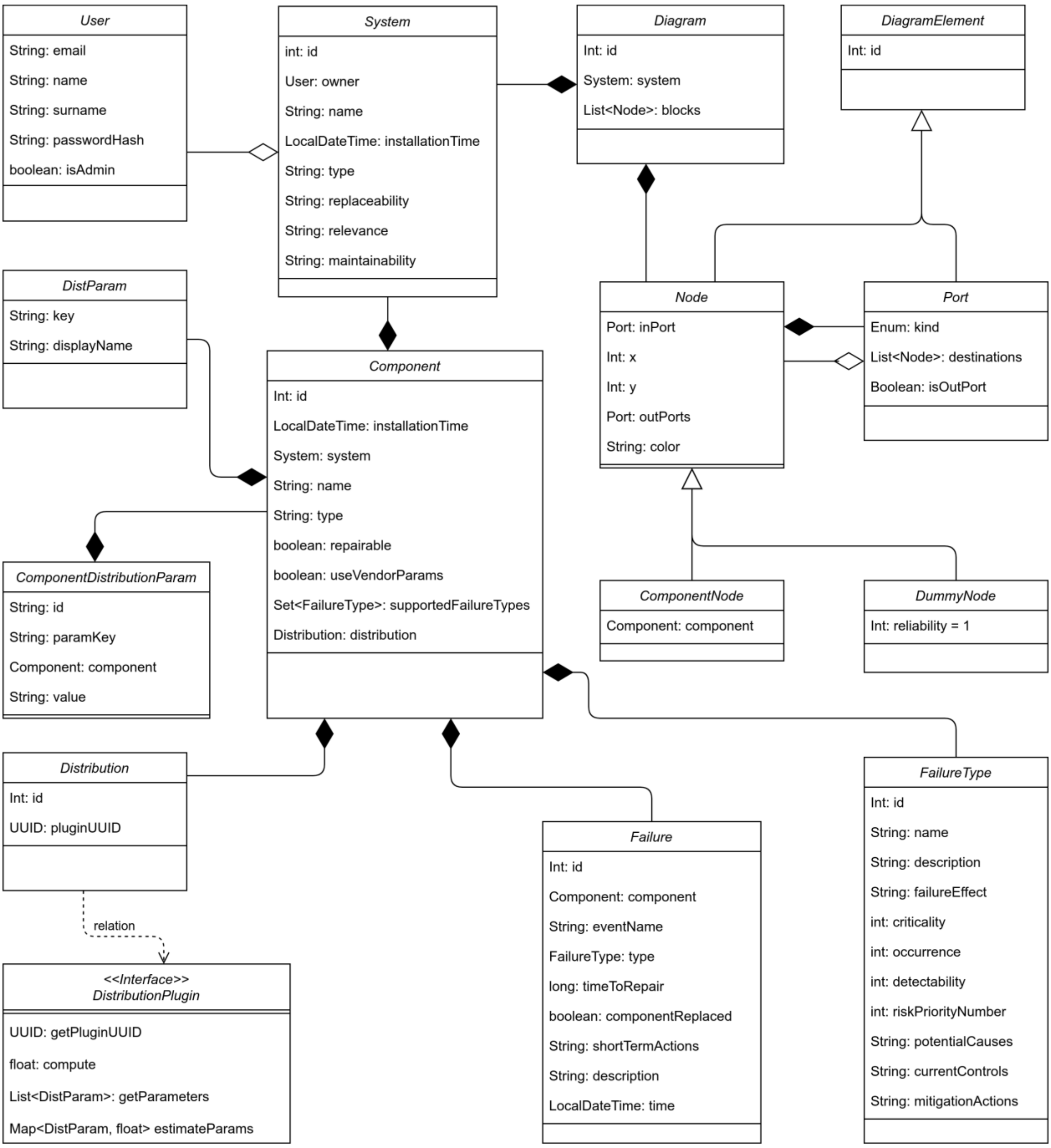


Figure 31 Data model

6.2.3.1 Class: System

This class aims to describe a system/machine.

Name	Type	Description
id	int	This attribute allows the platform to identify the system uniquely
owner	User	Owner of the system, user in which the system is related
name	String	Name of the system

installationTime	LocalDateTime	Installation date of the system, first start up
type	String	Type of system, a descriptive value
replaceability	int	Replaceability value of the system {1-3}
relevance	int	Relevance value of the system {1-3}
maintainability	int	Maintainability value of the system {1-3}

6.2.3.2 Class: Component

This class is meant to describe a component that is part of a system.

Name	Type	Description
id	int	This attribute allows the platform to identify the Component uniquely
intallationTime	LocalDateTime	The date in which the component has been installed in the system
system	System	The system to which the component belongs
name	String	The name of the component
type	String	Type of component, a descriptive value
repairable	Boolean	Identifies if the component is repairable
useVenrodParams	Boolean	Identifies if the component should use the vendor parameters or estimate them trough the distribution
supportedFailureTypes	Set<FailureType >	A set of FailureType associable to a Failure
distribution	Distribution	The distribution function that is used to estimate the parameters

6.2.3.3 Class: FailureType

This class describes a failure type/mode, which a component can generate.

Name	Type	Description
id	int	This attribute allows the platform to identify the FailureType uniquely
name	String	Name of the FailureType
description	String	Description of the type
failureEffect	String	The effect that derives from the failure
occurrence	int	The occurrence of the failure {1-3}
detectability	int	The detectability of the failure {1-3}
riskPriorityNumber	int	The Risk Priority Number of the failure {1-3}
potentialCauses	String	The potential causes of this kind of failure
currentControls	String	The current controls of this kind of failure
mitigationAction	String	The actions that are undertaken to mitigate the failure

6.2.3.4 Class: Failure

The aim of this class is to describe a failure event on a component.

Name	Type	Description

id	int	This attribute allows the platform to identify the Failure uniquely
component	Component	The component on which the failure has occurred
eventName	String	The name of the event that describes the failure
type	FailureType	The type of failure
timeToRepair	long	Time to repair the failure [seconds]
componentReplaced	boolean	Identifies if the component has been replaced
shortTermAction	String	The short term actions undertaken
description	String	The description of the failure
time	LocalDateTime	The date and time in which the failure has occurred

6.2.3.5 Diagram

The aim of this class is to assign a diagram (RBD model) to a system.

Name	Type	Description
id	int	This attribute allows the platform to identify the Diagram uniquely
system	System	The system which the diagram is part of

6.2.3.6 AbstractClass: DiagramElement

Abstract class that describes an element present on the diagram.

Name	Type	Description
id	int	This attribute allows the platform to identify the DiagramElement uniquely

6.2.3.7 Class: Node

The node is an actual entry on the diagram, which is composed by x and y coordinates and in/out ports. Since there can be multiple types of node, this class is defined as abstract.

Name	Type	Description
inPort	Port	The input port of the node
x	Int	The x position of the node
y	Int	The y position of the node
outPort	Port	The output port of the node
color	String	The node color

6.2.3.8 Class: Port

The aim of this class is to describe a port of a node. A port can be related to other ports.

Name	Type	Description

kind	Enum	The kind of output port {SEQUENTIAL, PARALLEL, ...}
destinations	List<Port>	List of destinations input ports
isOutPort	Boolean	Identifies if the port is an output

6.2.3.9 Class: ComponentNode

This class describes a component node, which allows the tool to calculate the system reliability.

Name	Type	Description
component	Component	The node related component

6.2.3.10 Class: DummyNode

This class has no attributes, this because this kind of object serves just as a note to start a new branch.

6.2.3.11 Distribution

A distribution param is a parameter that is required to calculate the reliability of a component trough a distribution.

Name	Type	Description
key	String	The key that identifies the parameter
displayName	String	The name of the parameter

6.2.3.12 Class: ComponentDistributionParam

The aim of this class is to declare the value of each parameter.

Name	Type	Description
id	String	This attribute allows the platform to identify the ComponentDistributionParam uniquely
paramKey	String	The key of the parameter
component	Component	The component to which the value is related
value	String	The value of the param

6.2.3.13 Class: Distribution

This class allows the system to describe a distribution.

Name	Type	Description
id	int	This attribute allows the platform to identify the distribution uniquely
pluginUUID	UUID	UUID of the plugin related to the distribution
name	String	Name of the distribution

6.2.3.14 Plugin

Interface: DistributionPlugin

This interface is used to implement new plugins.

Method Name	Return type	Description
getPluginUUID	UUID	This attribute allows the platform to identify the plugin uniquely, and establish a relationship with a Distribution
compute	float	Result of the computation
getParameters	List<DistParam>	Return the requested parameters
estimateParams	Map<DistParam, float>	Return a map of estimated parameters

6.3. User interface

The aim of the following sections is to describe how the user interface has been implemented. The aim of the GUI is to be as simple as possible, maintaining all the necessary functionalities reported by the requirements. To reduce eye fatigue, during work on the tool, a dark color palette has been chosen, as showed in Figure 32.

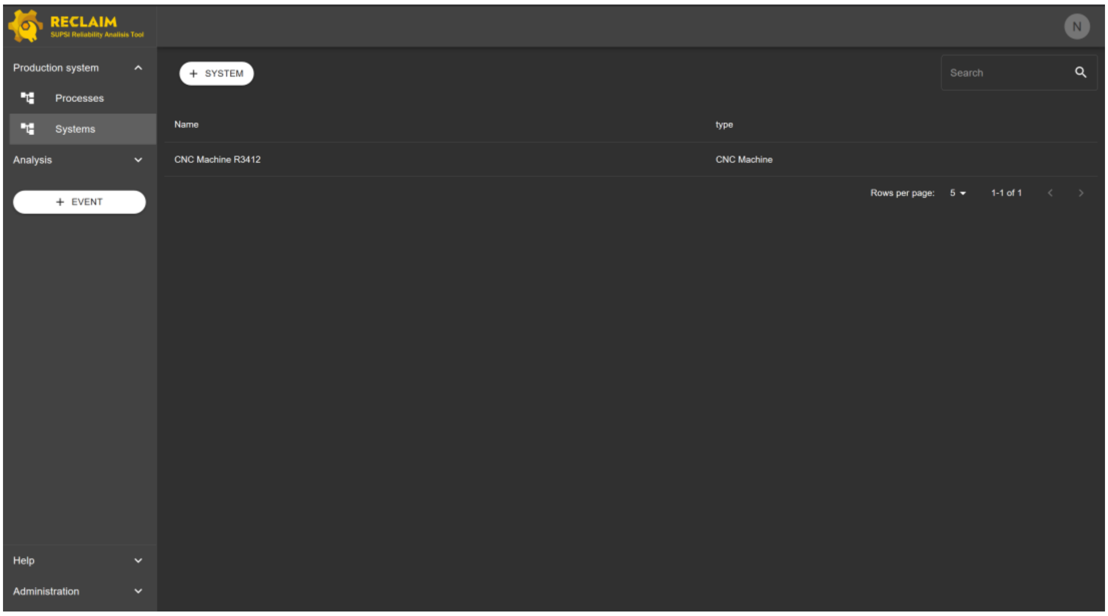


Figure 32 RAT: Systems' menu

To support the “newbie” users a series of guided procedures has been designed and implemented. These procedures are implemented on the tool for each entity creation and for each analysis (FMEA and RBD). Figure 33 shows the form used to implement the guided procedure to create a new component.

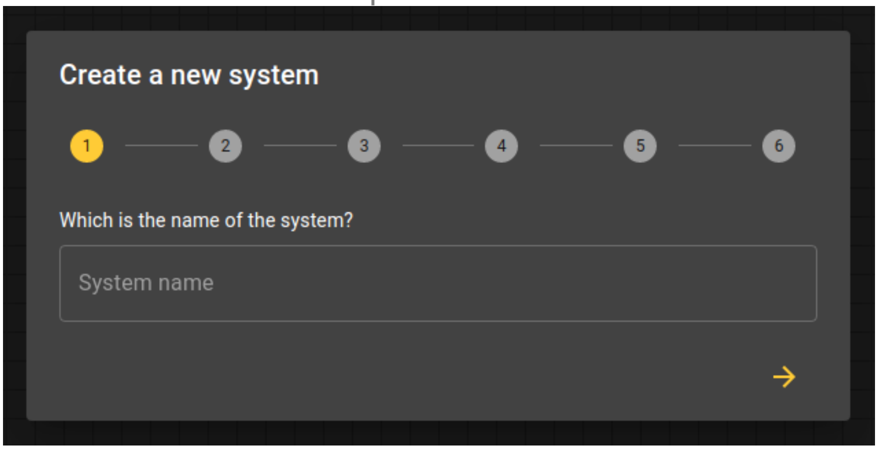
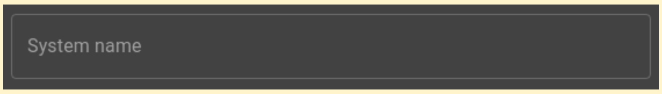
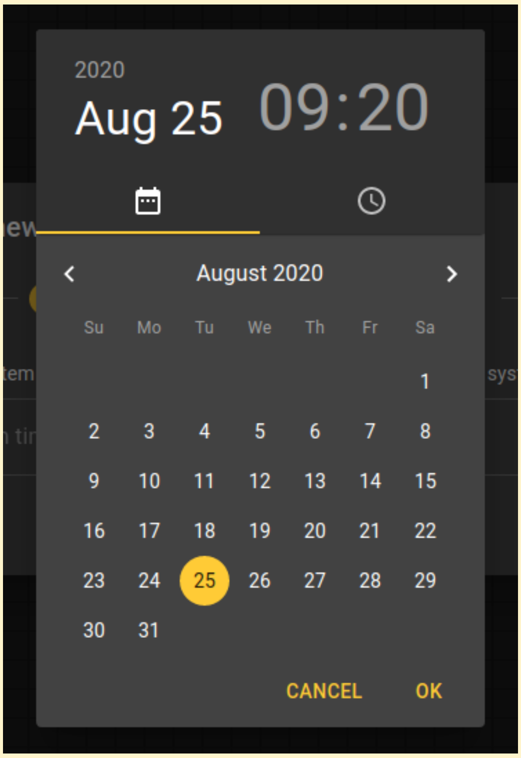


Figure 33 Example of guided procedure

To keep as much simplicity as possible 5 input GUI widgets have been defined:

Name	Type	Example
Text input	Text/number	
Slider	Number / range	
Calendar	Date (time)	

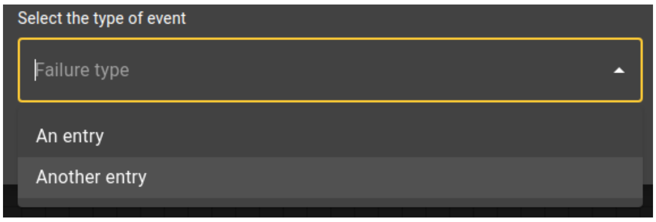
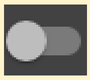
Drop-down menu	Entity	
Toggle	Boolean	

Table 12 Input widgets

Also the procedure navigation has been kept as much simple as possible by implementing 3 controls listed in Table 13:

Name	Behavior	Example
Backward button	Go to the previous step in the procedure	
Save button	Save the currently defined entry	
Forward button	Go to the next step in the procedure (a data check will be done on the input just inserted)	

Table 13 Navigation controls

The *forward button* and *backward button* are always visible, meanwhile the *save button* is visible only at the last step of the procedure. The following sections will illustrate the definitions of every step for each procedure. On the tables, the GUI widgets are referenced as follows:

I: text input
S: slider
C: calendar
D: drop-down menu
T: toggle

6.4. Deployment

The aim of this section is to explain how to deploy the application and start the latest version. To allow an easy deployment the main components of the tool have been containerized with the help of docker. Docker (www.docker.com) simplifies the process of create, deploy, and run applications by using containers, and containers allow a

developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.
Each component is container image and it is generated using a Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.

6.4.1 Back-end

The back-end docker file is composed of two steps:

- Build: the dependencies are resolved and the application is built into an executable file
- Execution: the build outcome (an executable file) is executed into a lightweight container.

The Docker file is composed as follows:

```
FROM maven:3.6.3-jdk-11 AS build
RUN mkdir -p /workspace
WORKDIR /workspace
COPY pom.xml /workspace
COPY src /workspace/src
RUN mvn clean package
FROM openjdk:12-alpine
COPY --from=build /workspace/target/*.jar app.jar
COPY plugins plugins
EXPOSE 8080
ENTRYPOINT ["java","-jar","app.jar"]
```

The built image is then pushed to the project docker repository to be available to everyone: *sps.registry.jetbrains.space/p/qu4lity/containers/rat_platform:latest*

6.4.2 Front-end

Similarly to the back-end docker file, the process to build an image of the front-end is subdivided into two distinct stages:

- Build: the dependencies are resolved and the application is built into a minified javascript file and html file.
- Execution: the build outcome (a folder) is exposed from a lightweight container.

The Docker file is composed as follows:

```
FROM node:14 as react-build
WORKDIR /app
COPY . ./
ADD .env-docker .env
RUN npm install --silent
RUN npm run build

FROM nginx:alpine
COPY nginx.conf /etc/nginx/conf.d/default.conf
```

```
COPY --from=react-build /app/build /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

The built image is then pushed to the project docker repository to be available to everyone: *sps.registry.jetbrains.space/p/qu4lity/containers/rat_client:latest*

6.4.3 Early access version

The fundamental requirement to run this application is to have docker installed, a fully comprehensive guide is available on the docker website at:
https://docs.docker.com/engine/install/
To have a ready to use environment is only necessary to run a docker-compose script, in which it will download all the necessary files and image from a remote server.
The *docker-compose.yaml* file is composed as follows:

```
version: '3'
services:

  rat-postgres:
    image: "postgres:9.6-alpine"
    container_name: rat-postgres
    volumes:
      - rat-data:/var/lib/postgresql/data
    ports:
      - 5432:5432
    environment:
      - POSTGRES_DB=rat_reclaim
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=postgres

  rat-platform:
    image: "sps.registry.jetbrains.space/p/qu4lity/containers/rat_platform:latest"
    container_name: rat-platform
    environment:
      - DB_SERVER=rat-postgres:5432
      - POSTGRES_DB=rat_reclaim
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=postgres
    ports:
      - 8080:8080
    links:
      - rat-postgres

  rat-client:
    image: "sps.registry.jetbrains.space/p/qu4lity/containers/rat_client:latest"
    container_name: rat-client
    ports:
      - 4200:80
```

```
    links:
      - rat-platform

volumes:
  rat-data:
```

For testing purposes this docker-compose has also a third component which is the database, it uses a default postgre image from the official docker repository. This allows the user to try the tool without the need to install additional software.

6.5. User Manual

User manual is integrated in the web-application. It is possible to access to the web-application at this link: http://isteps-sps-01.supsi.ch/reclaim/login

6.6. RAT release 1.0: features

The release 1.0 of the Reliability Analysis tool provides the following high-level functionalities:

- Registration of new users (account confirmation and password recovery are not currently available).
- Creation and characterisation of processes, systems, components, failure modes (FMEA) and events.
- RBD analysis through parallel (total redundancy) and series relations.

6.6.1 Hypothesis behind the RAT's RBD analysis

The following hypothesis are at the basis of the RBD analysis calculations:

- Failures are considered independent from system and component's history and are a consequence of purely random events.
- The approach proposed by the tool focuses on the useful life of the systems, leaving out the infant life and the end of the life. Considering all stages of the life cycle could generate inconsistent results.
- The tool currently analyses data based on the dates of installation, of repair and of occurrence of a failure event. It is assumed that the distribution of the workload during the year is constant. Future developments could foresee the possibility of including analysis and calculations based on working hours.
- A repairable component that is repaired is considered as "new". Future developments may include the possibility to include coefficients associated to component age or number of repairs.
- The reliability of a system often depends on the operating conditions in which it operates. Distribution fitting works if all the events added and the operating conditions of the system have not changed relevantly. If these have been altered, the approximations made by fitting the distributions may not be reliable.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 869884

7. Conclusion

The activities carried out in Task 2.5 allowed to design and develop a web-application supporting companies, in particular SMEs, which have lack of competences and no or weak approaches to maintenance and failure management, in starting their journey toward the digitalisation of assets health and maintenance management. In particular, analysing RECLAIM's use cases, the PODIUM one seems to be a perfect example where to apply and validate the RAT. The situation of PODIUM is different compared to the other User partners of RECLAIM. PODIUM does not have any machine connected to any device and/or software application to retrieve any kind of information from the machines. Every single process is manual, and they do not even have a formal schedule for checking the health of the machines. Despite a running process that will provide a digitalised background able to support the more automated acquisition of these information, the collection of data by means of a manual intervention is strategic to start having a collection of failures of the machines, enabling to create an historian to be used in the following development of more complex predictive algorithms. Such situation, beyond the RECLAIM demonstrators, is common to several micro and small enterprises that could benefit from this technology.

RAT can be used to start structuring failure modes and events and performing preliminary analysis on critical machines and equipment. To this end, the collection and structuring of data carried on by means of the RAT will be instrumental to support the filling out of the KPIs and the reliability metrics being designed in T3.2.

Task 2.5 continues until M30 of the project. This gives the opportunity to extend, improve and validate the RAT. In particular, the following features will be included in the next release:

- RBD analysis relying on parallel (partial redundancy), parallel (stand-by) and, eventually, parallel multi state (fractioning) relations;
- RBD analysis refining and improvement according to user feedbacks;
- Estimation of Weibull 2 param, Weibull 3 param, Exponential, Normal, Lognormal parameters based on failure events occurrence. Eventually also Gumbel, Rayleigh, Uniform will be included;
- Import and export of failure modes, events, systems and components;
- Statics analysis of failure events and production system entities;
- APIs to communicate with RECLAIM platform.

Finally, feedbacks will be collected from users in order to continuously improve usability and add any other relevant function.

References

ADL Group. (2020). *aldservice.com/*. Retrieved from https://aldservice.com/reliability-products/rams-software.html

Alsyouf, I. (2007). The role of maintenance in improving companies’ productivity and profitability. *International Journal of production economics, 105*(1), 70-78.

Arunraj, N. S., & Maiti, J. (2010). Risk-based maintenance policy selection using AHP and goal programming. *Safety science*, *48*(2), 238-247.

Baglee, D., Gorostegui, U., Jantunen, E., Sharma, P., & Campos, J. (2017). How can SMEs adopt a new method to advanced maintenance strategies? A Case study approach. *30th International Conference on Condition Monitoring and Diagnostic Engineering Management*, *COMADEM 2017*. Preston (UK).

Bevilacqua, M., & Braglia, M. (2000). The analytic hierarchy process applied to maintenance strategy selection. *Reliability Engineering & System Safety*, *70*(1), 71-83.

BQR. (2020). *www.bqr.com*. Retrieved from https://www.bqr.com/products/care/

Carlson, C. S. (2012). *Effective FMEAs*. John Wiley & Sons.

Chelidze, D., & Cusumano, J. (2004). A dynamical systems approach to failure prognosis. *Journal of Vibration and Acoustics*.

Christiansen, B. (2020). *A Ridiculously Simple Guide To Maintenance KPI*. Retrieved from LimbleCMMS: https://limblecmms.com/blog/maintenance-kpi/

Christiansen, B. (2020, 06 22). *MTTR, MTBF and MTTF: guide to failure metrics*. Retrieved from limblecmms.com: https://limblecmms.com/blog/mttr-mtbf-mttf-guide-to-failure-metrics/

Coleman, J. P. (2020, 06 16). *Reliability What is it?* Retrieved from maintworld.com: https://www.maintworld.com/Asset-Management/Reliability-What-Is-It

Confalonieri, M., Barni, A., Valente, A., Cinus, M., & Pedrazzoli, P. (2015). An AI based decision support system for preventive maintenance and production optimization in energy intensive manufacturing plants. *IEEE International Conference* .

Das, K., Lashkari, R. S., & Sengupta, S. (2007). Machine reliability and preventive maintenance planning for cellular manufacturing systems. *European Journal of Operational Research, 183*(1), 162-180.

Dong, M., & He, D. (2007). A segmental hidden semi-markov model (hsmm)-based diagnosticsand prognostics framework and methodology. *Mechanical Systems and Signal Processing*, 2248–2266.

Dr. Smith, D. J., & Simpson, K. G. (2016). Data, Failure Rate and Mode. In *The Safety Critical Systems Handbook (Fourth Edition)*. Sciencedirect.

Dynaway. (2019). *Failure Data Analysis in Maintenance (MTTR, PPC and more)*. Retrieved from www.dynaway.com: https://www.dynaway.com/blog/the-importance-of-maintenance-failure-metrics-as-kpis

Fault Tree Analysis. (n.d.). Retrieved from Six Sigma Study Guide: https://sixsigmastudyguide.com/fault-tree-analysis/

Guess, F., & Proschan, F. (1985). *Mean residual life: theory andapplications. Handbook of Statistics*. NewYork: 215–224.

H. Guo, S. W. (2009). Reliability analysis for wind turbines with incomplete failure data collected from after the date of initial installation. *Rel. Eng. Syst. Safety*, vol 93 - n6 - 1057 -1063.

H. Sutherland, T. R. (2003). Prognostics, A newlook at statistical life prediction for condition-basedmaintenance. *Proc. IEEE Aerospace Conf.*,, vol 7, 3131 - 3136.

HBM Prenscia. (2020). *Reliasoft.com*. Retrieved from https://www.reliasoft.com/products

Heng, A., Tan, A., Mathew, J., Montgomery, N., Banjevic, D., & Jardine, A. (2009). Intelligent condition-based prediction of machinery reliability. *Mechanical Systems and SignalProcessing, 23*(5), 1600 – 1614.

(1990). *IEC 60050 (191)*.

ISO/TR 12489. (2013). Reliability modelling and calculation of safety systems. . *Global Standards*.

JetBrains. (2020). *https://www.jetbrains.com/*.

K. Goode, J. M. (2000). Plant machinery working life prediction method utilizing reliability and condition-monitoring data. *Proc. Institut. Mechan. Engineers _ Process Mechanical Eng*, val 214 - 109-122.

Kendon, P. (2019, September 24). *How Well Do You Understand the Key Metrics for Reliability and Maintenance?* Retrieved from https://www.prometheusgroup.com/posts/how-well-do-you-understand-the-key-metrics-for-reliability-and-maintenance

Kumar, A., & Saini, M. (2014). Cost-benefit analysis of a single-unit system with preventive maintenance and Weibull distribution for failure and repair activities. *Journal of Applied Mathematics, Statistics and Informatics*, 5-19.

Labib, A. W. (2004). A decision analysis model for maintenance policy selection using a CMMS. *Journal of Quality in Maintenance Engineering*.

Lai, C. D., Murthy, D. N., & Xie, M. (2006). Weibull distributions and their applications. *Springer Handbook of Engineering Statistics,*, 63-78.

Le Son, K. F. (2013). Remaining useful life estimation based on stochastic deterioration models: A comparative study. . *Reliability Engineering & System Safety,*, 112, 165-175.

leanproduction.com. (2020). Retrieved August 12, 2020, from https://www.leanproduction.com/oee.htmlm

Liao, L., & Köttig, F. (2014). Review of Hybrid Prognostics Approaches for Remaining Useful Life Prediction of Engineered Systems, and an Application to Battery Life Prediction. *IEEE Trasactions on reliability*.

Love, C. E., & Guo, R. (1996). Utilizing Weibull failure rates in repair limit analysis for equipment replacement/preventive maintenance decisions. *Journal of the operational Research Society, 47*(11), 1366-1376.

Lundteigen, M. A., & Rausand, M. (n.d.). *Fault Tree Analysis (FTA)*. Retrieved from https://www.ntnu.edu: https://www.ntnu.edu/documents/624876/1277046207/SIS+book+-+chapter+05+-+Introduction+to+fault+trees/fa8ba01a-3baf-4bb8-94ed-116bf5bc6b44

Lundteigen, M. A., & Rausand, M. (n.d.). *Reliability Block Diagram (RBD)*. Retrieved from https://www.ntnu.edu: https://www.ntnu.edu/documents/624876/1277046207/SIS+book+-+chapter+05+-+Introduction+to+RBDs/61af88f9-b6d6-402a-94f7-902706c921c7

Luo, J., Pattipati, K., Qiao, L., & Chigusa, S. (2008). Model-based prognostic techniques ap-plied to a suspension system. *Transactions on Systems, Man, and Cybernetics*, 1156–1168.

Macchi, M., Garetti, M., Centrone, D., L., F., & Pavirani, G. P. (2012). “Maintenance management of railway infrastructures based on reliability analysis. *Reliability Engineering & System Safety, 104*, 71-83.

Mazhar, M., Kara, S., & Kaebernick, H. (2007). Remaining life estimation of used components in consumer products: Life cycle data analysis by Weibull and artificial neural networks. *Journal of Operations Management, 25 (2007), pp. 1184-1193*, 1184-1193.

Medjaher, K., Tobon-Mejia, D. A., & Zerhouni, N. (2012). Remaining Useful Life Estimation of Critical Components with Application to Bearings. *IEEE Journal of Transactions on Reliability, 61*(2), 292-302.

Moubray, J. (2001). *Reliability-centered maintenance*. Industrial Press Inc.,.

Nowlan, F. S., & Heap, H. F. (1978). *Reliability-centered maintenance*. United Air Lines Inc San Francisco Ca.

Pobočíková, I., & Sedliačková, Z. (2014). Comparison of four methods for estimating the Weibull distribution parameters. *Applied Mathematical Sciences*, 4137-4149.

Raza, M. (2020, 06 15). *System Reliability and Availability Calculations*. Retrieved from bmc.com: https://www.bmc.com/blogs/system-reliability-availability-calculations/

Relyence Corporation. (2020). *Relyence.com*. Retrieved from https://www.relyence.com/products/

Schenkelberd, F. (2020, 04 03). *Reliability Function*. Retrieved from accendoreliability.com: https://accendoreliability.com/reliability-function/

Shalabi, F., & Turkan, Y. (2017). IFC BIM-based facility management approach to optimize data collection for corrective maintenance. *Journal of performance of constructed facilities, 31*(1).

Soualhi, A., & Guy, C. (2019, August). Data driven methods for the prediction of failures. *researchgate*.

Stamatis, D. H. (2003). Failure mode and effect analysis: FMEA from theory to execution. *Quality Press*.

Stamatis, D. H. (2003). *Failure Mode and Effect Analysis: FMEA from Theory to Execution*.

Tahir, Z., Prabuwono, A. S., & Aboobaider, B. M. (2008). Maintenance decision support system in small and medium industries: an approach to new optimization model. International. *Journal of Computer Science and Network Security, 8*(11), 155-162.

Vesely, W. E., & Roberts, N. H. (1981). *Fault Tree Handbook* (Vol. 88). Systems and Reliability Research, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission.

Wang, L., Chu, J., & Wu, J. (2007). Selection of optimum maintenance strategies based on a fuzzy analytic hierarchy process. *International journal of production economics, 107*(1), 151-163.

www.mathworks.com. (2020). *RUL Estimation Using RUL Estimator Models*. Retrieved August 13, 2020, from https://www.mathworks.com/help/predmaint/ug/rul-estimation-using-rul-estimator-models.html

Xiongzi, C., Jinsong, Y., Diyin, T., & Yingxun, W. (2011). Remaining useful life prognostic estimation for aircraft subsystems or components: A review. *10th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*.

Yang, F., Habibullah, M. S., Zhang, T., Xu, Z., Lim, P., & Nadarajan, S. (2016). Health Index-based Prognostics for Remaining Useful Life Predictions in Electrical Machines. *IEEE Transactions on Industrial Electronics,, 63*(4), 2633-2644.